GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION
SPONSORED PROJECT INITIATION

Date: January 29, 1981

Project Title: Switched Capacitor Filter Design

Project No: E-21-643

Project Director: Dr. J. A. Connelly

Sponsor: Honeywell, Inc.; Ordnance Operations; Defense Systems Division;
600 Second Street, N. E.; Hopkins, Minnesota 55343

Agreement Period: From 1/1/81 Until ~~6/15/81~~ (R&D Perf. & Rpt.)

Type Agreement: Subcontract No. 336915, Research Project No. D053/AA

Amount:
$30,134 E-21-643
1,700 E-21-360
$31,834 TOTAL

Reports Required: Monthly Status Reports; Design Reviews; Final Report

Sponsor Contact Person (s):

Technical Matters

Mr. R. H. Cole
Mail Station MN 11-2100
Section Head Mines and Fusing
Honeywell, Incorporated
Ordinance Operations
Defense Systems Division
600 Second Street, N. E.
Hopkins, Minnesota 55343
612/931-6887

Contractual Matters
(thru OCA)

Mary E. Keller
M. T. Garten
Subsystems Administrator
Subsystems Procurement
Honeywell, Incorporated
Ordinance Operations
Defense Systems Division
600 Second Street, N. E.
Hopkins, Minnesota 55343
612/931-5899

Defense Priority Rating: None

Assigned to: Electrical Engineering (School/Laboratory)

COPIES TO:

- Project Director
- Division Chief (EES)
- School/Laboratory Director
- Dean/Director—EES
- Accounting Office
- Procurement Office
- Security Coordinator (OCA)
- Reports Coordinator (OCA)
- Library, Technical Reports Section
- EES Information Office
- EES Reports & Procedures
- Project File (OCA)
- Project Code (GTRI)
- Other Mr. C. E. Smith

25281/38

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION

Date: 9/1/81

Project Title: Switched Capacitor Filter Design

Project No: E-21-643

Project Director: Dr. J. A. Connelly

Sponsor: Honeywell, Inc.

Effective Termination Date: 7/31/81

Clearance of Accounting Charges: 7/31/81

Grant/Contract Closeout Actions Remaining:

x Final Invoice and Closing Documents
_ Final Fiscal Report
_ Final Report of Inventions
x Govt. Property Inventory & Related Certificate
_ Classified Material Certificate
_ Other ______

Assigned to: Electrical Engineering (School/~~Laboratory~~)

COPIES TO:

Administrative Coordinator
Research Property Management
Accounting Office
Procurement Office
Research Security Services
Reports Coordinator (OCA)
Legal Services (OCA)
Library, Technical Reports
EES Research Public Relations (2)
Project File (OCA)
Other: ______

E-21-672

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894- 2911

MEMORANDUM TO: Nancy S. McHan

FROM: J. A. Connelly - Project Director JAC.

SUBJECT: Monthly Status Report for Switched Capacitor Filter Design - Honeywell, Inc., Subcontract No. 336915 - Research Project No. E-21-643

DATE: February 10, 1981

The Project Director and Graduate Research Assistant, Mr. William Goolsby, visited Honeywell Defense Systems Division in Hopkins, Minnesota on January 7, 1981. A meeting was held with Bob Payne, John Timmerman, and Ken Ceola of Honeywell and Al Connelly and Bill Goolsby of Georgia Tech. The schedule and statement of work as presented in the original proposal to Honeywell of October 6, 1980 was reviewed and modified. The modified tasks and schedule are included as a part of this report.

During the afternoon of January 7, 1981, John Timmerman, Ken Ceola, Bill Goolsby, and Al Connelly visited SSEC and met with Jim Gorecki. A discussion was held concerning the practical limitations and progress toward the implementation of the notch and bandpass filters designed for the NEARTIP application.

Work by Georgia Tech during this activity period has focused on the development and implementation of a FORTRAN computer program for the Honeywell system to aid in the design of the following switched capacitor filters:

Notch (second order with finite rejection)
Bandpass (second order)
Lowpass (second order for Butterworth and Chebychev responses).

Consideration has been given to selecting appropriated SCF circuit configurations which minimize parasitic effects. To date FORTRAN programs have been written and installed on Honeywell's Computer System for the Notch and Bandpass filters. These programs correct for warping effects introduced by sampled data approach and determine all required capacitor values. Current efforts are directed toward providing the same computer analysis routine for the lowpass switched capacitor filter.

Revised Schedule and Statement of Work
February 9, 1981

1. Identification of the specific types of filters to be developed. Travel to the DSD Facility in Hopkins, MN is anticipated for the Principal Investigator and the Graduate Research Assistant. This task to be completed by January 15, 1981. (completed)

2. Selection of appropriate switched capacitor circuit configurations to minimize parasitic effects. These will include a second-order notch filter with finite rejection, a second-order bandpass filter, and a second-order low pass filter for Butterworth and Chebychev responses. The biquad versus ladder and ladder-leapfrog configurations will be studied with respect to sensitivity to capacitor matching tolerances and to the total required chip area. To be completed by March 15, 1981.

3. Development and installation on Honeywell's Computer facilities of Computer Aid Design Programs for the filters described in (2) above. Documentation in the final report will be provided showing techniques for transforming design parameters into required component values through the use of the S- and Z- operators. To be completed by April 15, 1981.

4. Presentation of preliminary results to DSD personnel. On site inspection by Honeywell Personnel at Georgia Tech is anticipated. To be completed by April 15, 1981.

5. Investigation of available SCF computer simulation programs. Comparisons of new routines with DINAP II and DIANA. No substantial attempt is anticipated to modify the DINAP II routine to accommodate non-ideal operational amplifiers. To be completed by May 1, 1981.

6. Identification of guidelines for element layout and placement on the LSI chip. A review of the available literature including notes from the UCLA short course on SCF design is anticipated with significant articles and papers being noted. To be completed by June 1, 1981.

7. Preparation and completion of the Final Report on Switch Capacitor Filter Design. Travel to DSD by the Principal Investigator and Graduate Research Assistant for a design review is anticipated. To be completed by June 15, 1981.

A1-672

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894- 2911

MEMORANDUM TO: Nancy S. McHan

FROM: J.A. Connelly - Project Director

SUBJECT: Monthly Status Report for Switched Capacitor Filter Design - Honeywell, Inc., Subcontract No. 336915 - Research Project No. E-21-643

DATE: March 6, 1981

During the February activity period, work has focused upon the development and implementation of FORTRAN computer aided design (CAD) programs for second-order bandpass (BP) and lowpass (LP) filters. Specifically a design procedure has been developed for an ideal bandpass filter employing switched-capacitors and patterened after that of Martin [1]. This is a different bandpass configuration than that reported previously. The Martin state-variable filter produces an ideal bandpass response whereas the previous BP filter, patterened after Gregorian [2], functions as a low-pass filter with heavy peaking in the pass band.

A CAD procedure has been written for the Martin BP filter and installed on the Georgia Tech computer facilities. The program has been debugged and sample designs tested and verified. A notable feature of this CAD program is that it automatically corrects for element warping due to finite clock frequencies.

The Martin BP circuit has a second output port where a second-order lowpass response can be obtained. A CAD program has been written for the LP filter, installed, and tested on Georgia Tech's computer facilities. Specifically, this program allows the designer to select either a Butterworth or Chebychev response. If a Chebychev type is chosen, the designer then specifies the permitted ripple in the passband and the frequency where the response leaves the passband. This CAD routine also automatically corrects for element warping.

Work has been initiated to study ladder networks for SCF implementation. A comparison of the characteristics, complexities, and sensitivities of ladder configurations versus state-variable and bi-quad configurations will be the focus of the current activity period. Also the Martin bandpass and lowpass CAD programs will be transferred to the Honeywell computer system during March.

1. K, Martin, "Improved Circuits for the Realization of Switched-Capacitor Filters," IEEE Transactions on Circuits & Systems, Vol. CAS-27, No. 4, April 1980, pp. 237-241.
2. R. Gregorian, "Switched Capacitor Filter Design Using Cascaded Stages," IEEE Transactions on Circuits & Systems, Vol. CAS-27, No. 6, June 1980, pp. 515-521.

E-21-643

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894- 2911

April 9, 1981

MEMORANDUM

TO: Nancy S. McHan

FROM: J. A. Connelly - Project Director J.A.C.

SUBJECT: Monthly Status Report for Switched Capacitor Filter Design - Honeywell, Inc., Subcontract No. 336915 - Research Project No. E-21-643

Activity during March focused on developing ladder networks for switched-capacitor filter implementation. A design approach was proposed and refined whereby a second-order notch filter can be developed from a first-order low-pass ladder network. Appropriate equations describing this network were generated in both the S and Z - domains. Work is continuing for verifying the results using the DINAP analysis routine.

Consideration has been given to a new approach for reducing drastically the capacitor ratios previously required for implementation of the notch and bandpass filters for NEARTIP. Preliminary results indicate that maximum capacitor ratios of 50:1 are possible instead of the previous 260:1 values. Further work in this direction will continue during April.

The Martin bandpass and low pass computer aided design programs reported in the last Status Report have now been installed on Honeywell's computer system and are operational.

JAC/md

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

'ELEPHONE: (404) 894-2961

May 12, 1981

Mary E. Keller
Mail Station MN 11-2610
Subsystems Administrator
Subsystems Procurement
Honeywell, Incorporated
Ordinance Operations
Defense Systems Division
600 Second Street, NE
Hopkins, MN 55343

Dear Ms. Keller:

Enclosed please find the Monthly Status Report for the month of April, 1981.

Should you have any questions, please do not hesitate to contact me at (404) 894-2961.

Sincerely,

Roberta J. Barker
Electrical Engineering

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

TELEPHONE: (404) 894- 2911

May 11, 1981

MEMORANDUM

TO: Leamon Scott

FROM: J. A. Connelly - Project Director JAC

SUBJECT: Monthly Status Report for Switched Capacitor Filter Design - Honeywell, Inc., Subcontract No. 336915 - Research Project No. E-21-643

On April 22, 1981 a mid-term design review was held at Georgia Tech. Mr. John Timmerman from Honeywell reviewed progress made on the project with J.A. Connelly and W.N. Goolsby. Discussions and demonstrations were directed toward the computer-aided design programs developed for low-pass, band-pass, and notch filters. Mr. Timmerman made several suggestions to improve the usefulness of these programs. Modifications are being made currently to include these program refinements.

Mr. Goolsby and Dr. Connelly attended the International Symposium on Circuits and Systems held in Chicago on April 27-29, 1981. Approximately thirty papers were presented on switched capacitor research by the leading authorities from throughout the world. Of particular promise is the possibility of obtaining a new computer analysis program called SCANAL from the Swiss Federal Republic in Zurich. This program permits modeling switched-capacitor ladder networks with arbitrary clock phasing and finite switch parameters. Negotiations have begun to determine the possibility of securing a copy of SCANAL.

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING
ATLANTA, GEORGIA 30332

LEPHONE: (404) 894-2911

August 7, 1981

Mr. Leamon R. Scott
Georgia Institute of Technology
Contract Administration
Administration Building
Atlanta, Georgia 30332

Dear Mr. Scott:

This is to inform you that all research activities associated with the Switched Capacitor Filter Design Contract between Georgia Tech and Honeywell, Inc. have been concluded. The approved information is:

Sponsor:	Honeywell, Inc. Defense Systems Division 600 2nd Street N.E. Hopkins, MN 55343
Title:	Switched Capacitor Filter Design
SubContract No.:	336915
Research Project No.:	E-21-643

Four copies of the Final Report were delivered to the Sponsor on August 3, 1981 and are now in the possession of R.L. Payne, K.D. Ceola and J.T. Timmerman, all of Honeywell, Inc.

Thank you for your assistance in the administration of this contract.

Sincerely

J.A. Connelly
Professor

JAC/jd

cc: Roberta Barker, E.E.
Marsha Seagraves, E.E.

E-21-643

SWITCHED-CAPACITOR FILTER DESIGN

by

J. A. Connelly

and

W. N. Goolsby

June 1981

Honeywell, Inc.
Defense Systems Division
Hopkins, Minnesota

Final Technical Report

TABLE OF CONTENTS

Abstract

This report is intended as an introduction to the design of switched-capacitor filters. Basic switched capacitor concepts are presented for simulating resistances. The effects of aliasing are discussed and two basic parasitic insensitive integrators are presented. Techniques for adopting analog active filters to SCFs are given. These techiques include derivation of the Z-transformed transfer function and creation of the SCF circuit. Computer-aided design programs are presented to enable the rapid design of second-order low-pass Butterworth, Chebyshev, and Generalized filters. CAD programs for second-order bandpass and band reject filters are presented also. These CAD programs correct for frequency warping due to finite clocking frequencies. Also the CAD programs permit the designer to easily minimize capacitance ratios and total circuit capacitance. Practical design examples are illustrated for using these CAD programs. A technique for synthesizing ladder filters is given, and an example is developed for generating a second order notch filter.

I. INTRODUCTION

The need for Switched-Capacitor Filters (SCF) arises because the largest RC time constant realizable with the usual on-chip resistors and capacitors is approximately 10 microseconds [1]. This is too small for audio band filters which can require time constants as long as 100 milliseconds. Standard IC technology can readily fabricate capacitors from about 1 picofarad to 100 picofarads with smaller values obviously being preferred since they require less chip area.* To achieve 1 to 100 ms time constants, resistors from 10 to 1000 megohms are needed. However, no integrated resistors this large can be diffused, implanted, or deposited with the accuracy and stability normally required. For these reasons, analog signal filtering in the audio frequency range normally has utilized external resistors and capacitors in hybrid IC structures. When accuracies of 1% or less are required, laser trimming is often the only way to realize the desired performance.

Switched capacitor filter design is based on simulating a resistance by using a storage capacitor and a two position (SPDT) switch, normally implemented with small MOSFETs. The SCF scheme lends itself well to ICs. Signal voltages can be sensed with FET input amplifiers. Op amps in NMOS and CMOS have already been demonstrated.

---

*In MOS technology, typical capacitance density ranges between 0.1 to 0.5 pF/square mil.

II. BASIC SWITCHED CAPACITOR CONCEPTS

A. Switched Capacitor Simulated Resistor

The basic SCF concept for simulating a resistor is shown in Fig. 2.1. At time t=0, the capacitor C is charged to a potential of V_1 volts, representing a stored charge of $Q = CV_1$. At $t=T_C$ switch S changes, and the voltage on C becomes V_2. Assuming $V_2 < V_1$, there is now less charge stored on C. The net charge transferred during the T_C second interval is

$$\Delta Q = C(V_1 - V_2). \tag{2-1}$$

When the switching cycle becomes repetitive with a period of T_C seconds, the rate of charge transfer, or current, is

$$i = \frac{\Delta Q}{T_C} = \frac{C(V_1 - V_2)}{T_C} \tag{2-2}$$

The simulated resistance is

$$R = \frac{V_1 - V_2}{i} = \frac{T_C}{C} = \frac{1}{f_c C} \tag{2-3}$$

where f_c is the repetitive clock cycle. As a practical numerical example; a 1.0 pF capacitor switched at 100 kHz frequency simulates a 10 MΩ resistance.

When the switching frequency is much larger than the signal frequencies of interest, the sampling time of the signal can be ignored and the switched capacitor can be considered as a direct replacement for a conventional resistor. As the signal frequency approaches the switching

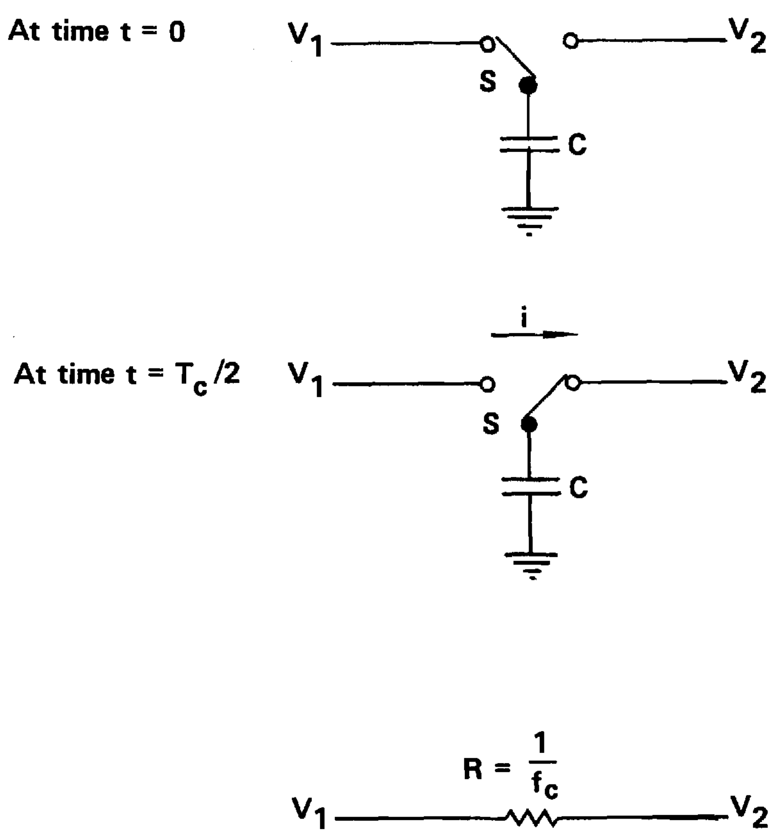


Fig. 2.1 A switched–capacitor equivalent of a resistor.

frequency, sampled-data techniques are required for proper analysis [2]. In any case the Nyquist criterion must be satisfied by keeping the highest signal frequency component of interest less than one-half of the clock frequency.

B. Implementation of the Switched Capacitor Resistor

The most common way of implementing the switched capacitor is to use MOS transistors and a two-phase clocking scheme as shown in Figure 2.2. The MOSFETs operate as switches controlled by nonoverlapping clock phases, ϕ_1 and ϕ_2, and a frequency $f_c = 1/T_C$. When ϕ_1 is high, Q_1 conducts charging C to V_1. When ϕ_2 is high, C is connected to V_2. Otherwise C is connected to neither V_1 nor V_2.

The principal reason why SCFs are so useful for accurate filtering in the audio band can be seen by examination of the basic analog integrator and its switched-capacitor dual shown in Fig. 2.3. The time constant for the analog integrator is $\tau = R_1C_2$ and for the SCF integrator,

$$\tau = T_C(C_2/C_1) \qquad (2\text{-}4)$$

The relative values of C_1 and C_2 are determined by photolithographic methods and matching accuracies of 0.1 percent have been demonstrated [3]. Furthermore, the MOS capacitor closely approaches the ideal capacitor characteristic and much better stability and linearity are obtained than that possible with diffused resistors. Finally, the capacitance ratio has very little temperature dependence. The dominant conclusion is that a switched-capacitor resistor makes possible the design of precise, stable, active RC audio band filters which can be fully integrated using MOS technology [4].

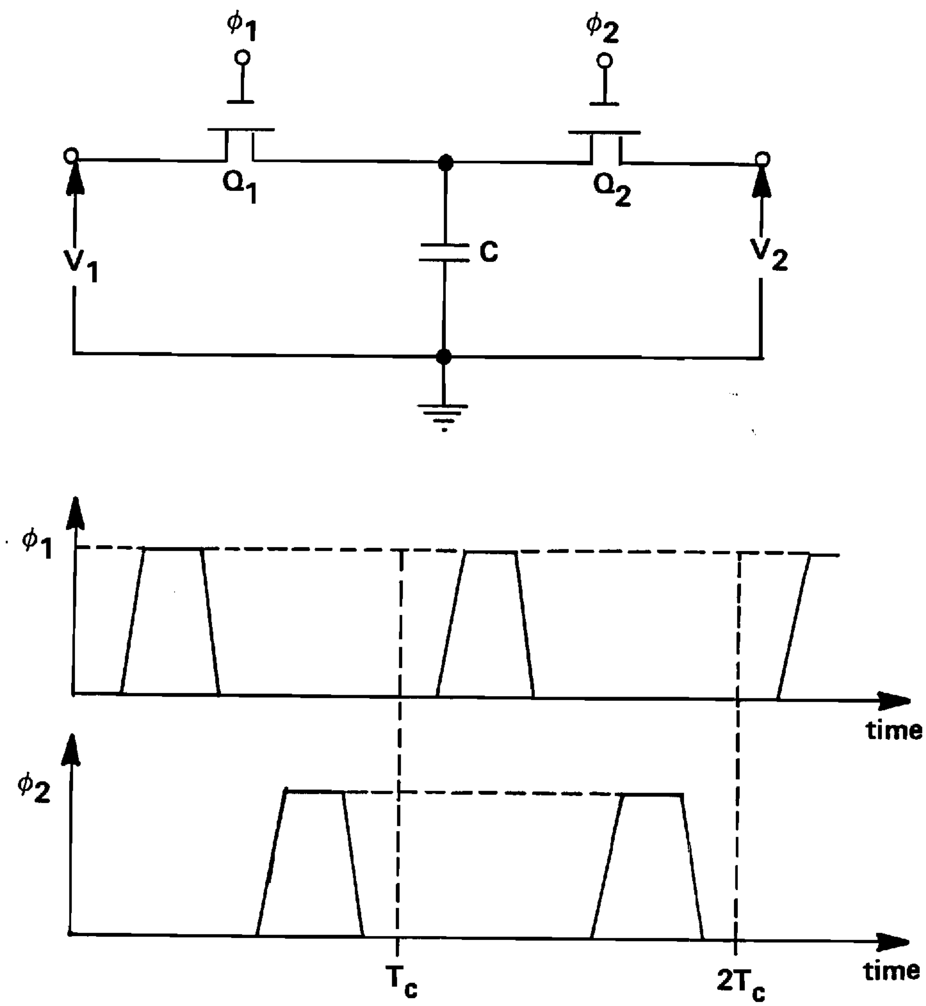


Fig. 2.2 An MOS implementation of a switched capacitor

C. Example of SCF Integrator

The inverting integrator of Fig. 2.3(a) is the basic circuit used in many analog active filters. Consider that the switch is in position 1 at some arbitrary time $(n-1)T_C$. The charge stored on C_1 is

$$Q_1\left[(n-1)T_C\right] = C_1\ V_{in}\left[(n-1)T_C\right] \tag{2-5}$$

The charge stored on C_2 is

$$Q_2\left[(n-1)T_C\right] = C_2\left[V_{out}\ (n-1)T_C\right] \tag{2-6}$$

One full cycle later at time nT_C, the previous charge on C_1 has been transferred to C_2 so that

$$Q_2(nT_C) = Q_2\left[(n-1)T_C\right] - Q_1\left[(n-1)T_C\right] \tag{2-7}$$

Therefore, we can write

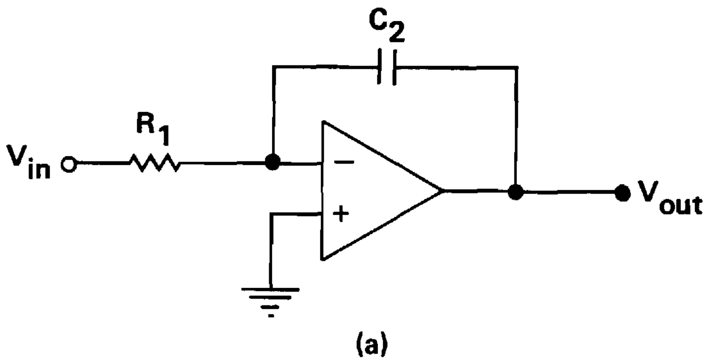
$$C_2\ V_{out}(nT_C) = C_2\ V_{out}\left[(n-1)T_C\right] - C_1\ V_{in}\left[(n-1)T_C\right] \tag{2-8}$$

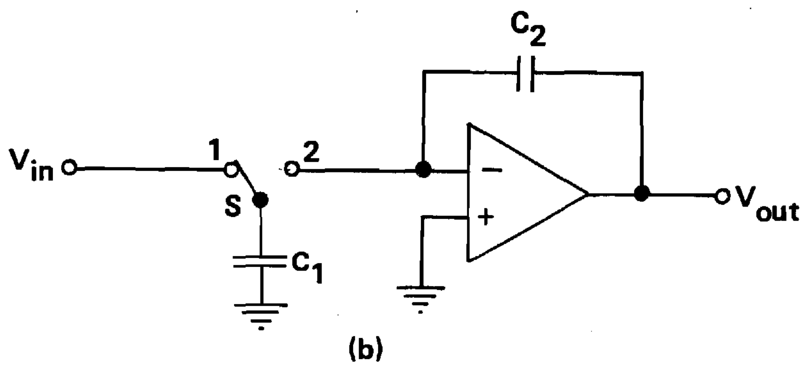
Taking the Laplace transform

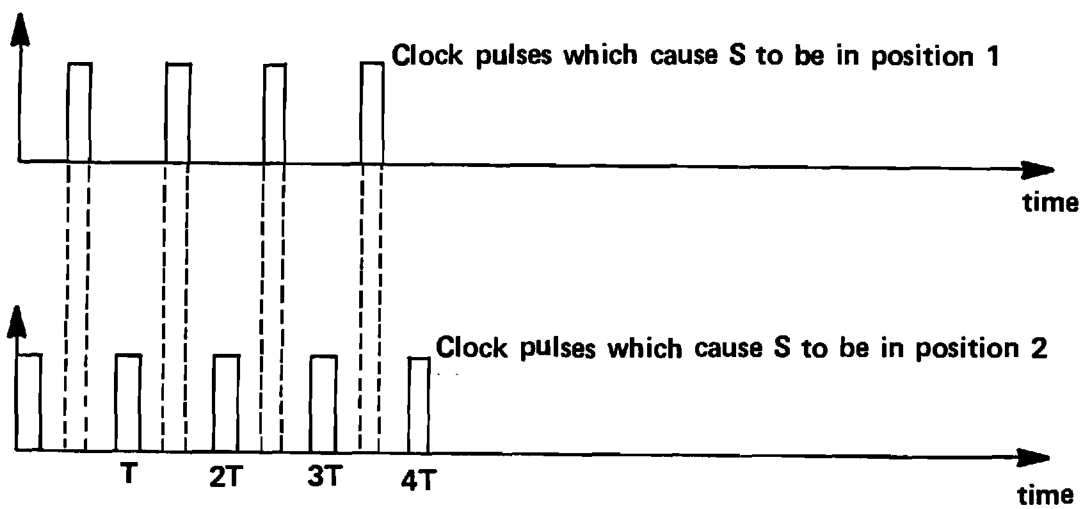
$$C_2 V_{out} e^{snT_C} = C_2 V_{out} e^{s(n-1)T_C} - C_1\ V_{in} e^{s(n-1)T_C} \tag{2-9}$$

After simplification we have

$$\frac{V_{out}}{V_{in}} = -\frac{C_1/C_2}{e^{sT_C} - 1}\ . \tag{2-10}$$







(c)

Fig. 2.3 Ideal integrator: (a) analog circuit; (b) SCF equivalent; and (c) two phase clock for driving switch S.

Recall that

$$e^{sT_C} = e^{j\omega T_C} = \cos \omega T_C + j \sin \omega T_C \qquad (2\text{-}11)$$

For $\omega T_C << 1$, Eq. 2-10 becomes

$$\frac{V_{out}}{V_{in}} \approx - \frac{C_1/C_2}{j\omega T_C} \qquad (2\text{-}12)$$

Thus, the SCF circuit of Fig. 2.3(b) approximates the ideal integrator as long as the product of the sampling period (T_C) and the highest signal frequency of interest (ω) is much, much less than 1.

As a practical example consider a sinusoidal input signal to the sampled-data integrator as shown in Fig. 2.4. Let the clock frequency be ten times the input signal frequency. Also, assume the peak amplitude of the input is 1.0 volts. The input signal is sampled in the middle of every clock cycle, and this value of V_{in} is held on C_1 for half a clock period until it is transferred to C_2. The output is held for a clock period until a new sample is transferred to the capacitor C_2. From Eq. 2-12,

$$V_{out} \approx - \frac{(C_1/C_2)(10f)(1 \text{ volt})}{j\ 2\pi f} = - \frac{5(C_1/C_2)}{j\pi} \text{ volts} \qquad (2\text{-}13)$$

Now, if we choose $C_1/C_2 = \pi/5$, the output voltage will have a 1 volt peak amplitude and will lag the phase of the input signal by 90 degrees as shown in Fig. 2.4(d). If we arbitrarily let the output voltage be unity at t=0, the value of V_{out} at $t=T_C$ can be found from Eq. 2-8 as

$$C_2\ V_{out}(T_C) = C_2(1 \text{ volt}) - C_1 \sin 18^o \qquad (2\text{-}14)$$

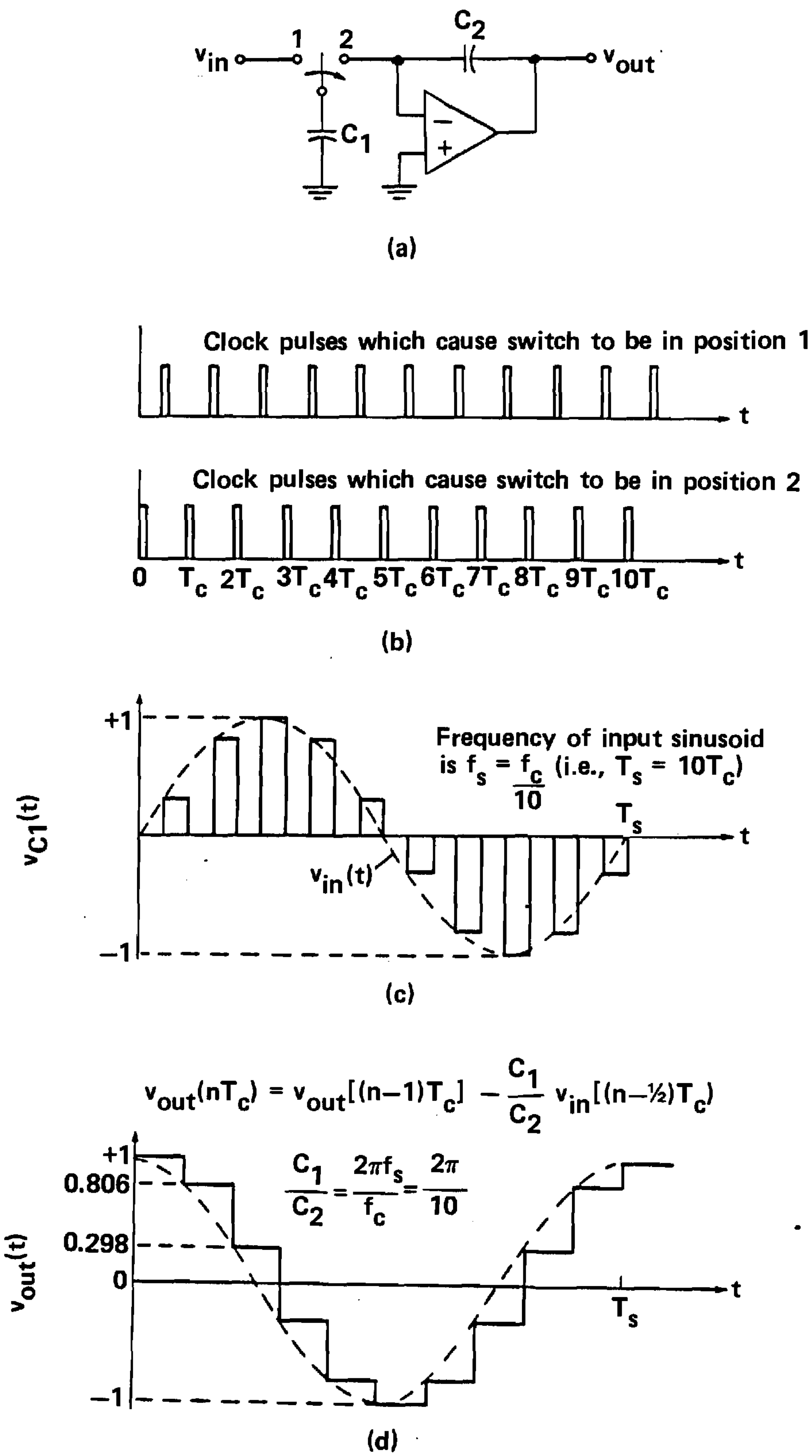


Fig. 2.4 Example of sampled-data integrator: (a) circuit; (b) clocking phases; (c) input signal and voltage stored on C_1; and (d) output voltage.

$$V_{out}(T_C) = 1 \text{ volt} - \frac{C_1}{C_2}(0.309) = 0.806 \text{ volts} \tag{2-15}$$

This technique can be repeated for the other nine samples per period of the input signal to determine the V_{out} at each interval. As the input frequency is made much, much less than the sampling frequency, the output distortion is reduced.

D. The Effect of Aliasing in Sampled-data Filters

The previous example and Eqs. 2-10 through 2-12 demonstrate how closely the sampled-data integrator approximates the ideal analog integrator. This distinction between an analog filter and a sampled-data filter is further illustrated in Fig. 2.5. As the signal frequency f_s approaches the clocking frequency, the magnitude of the frequency response increases rather than going to zero as in pure analog filters. This effect is called aliasing. It is often necessary to use an additional analog filter in cascade with the sampled-data filter to correct for aliasing effects.

E. Parasitic Insensitive Integrators

The inverting integrator of Fig. 2.4 is sensitive to parasitic effects [5]. This results from additional parasitic capacitance between the upper plate of C_1 and ground. The parasitic capacitance increases the effective value of C_1, thereby changing the critical C_1/C_2 design ratio.

To avoid the inaccuracies and nonlinear effects due to parasitic capacitive effects, two improved integrator circuits have been developed and are shown in Fig. 2.6 [6]. The transfer function of these two integrators are completely independent of the stray capacitance between any node

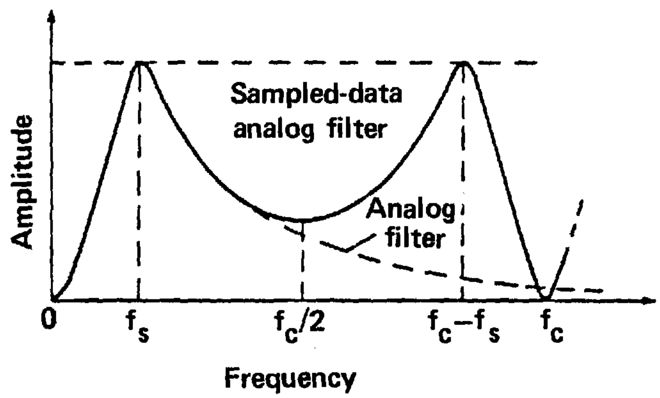


Fig. 2.5 The effect of aliasing in sampled-data filters.

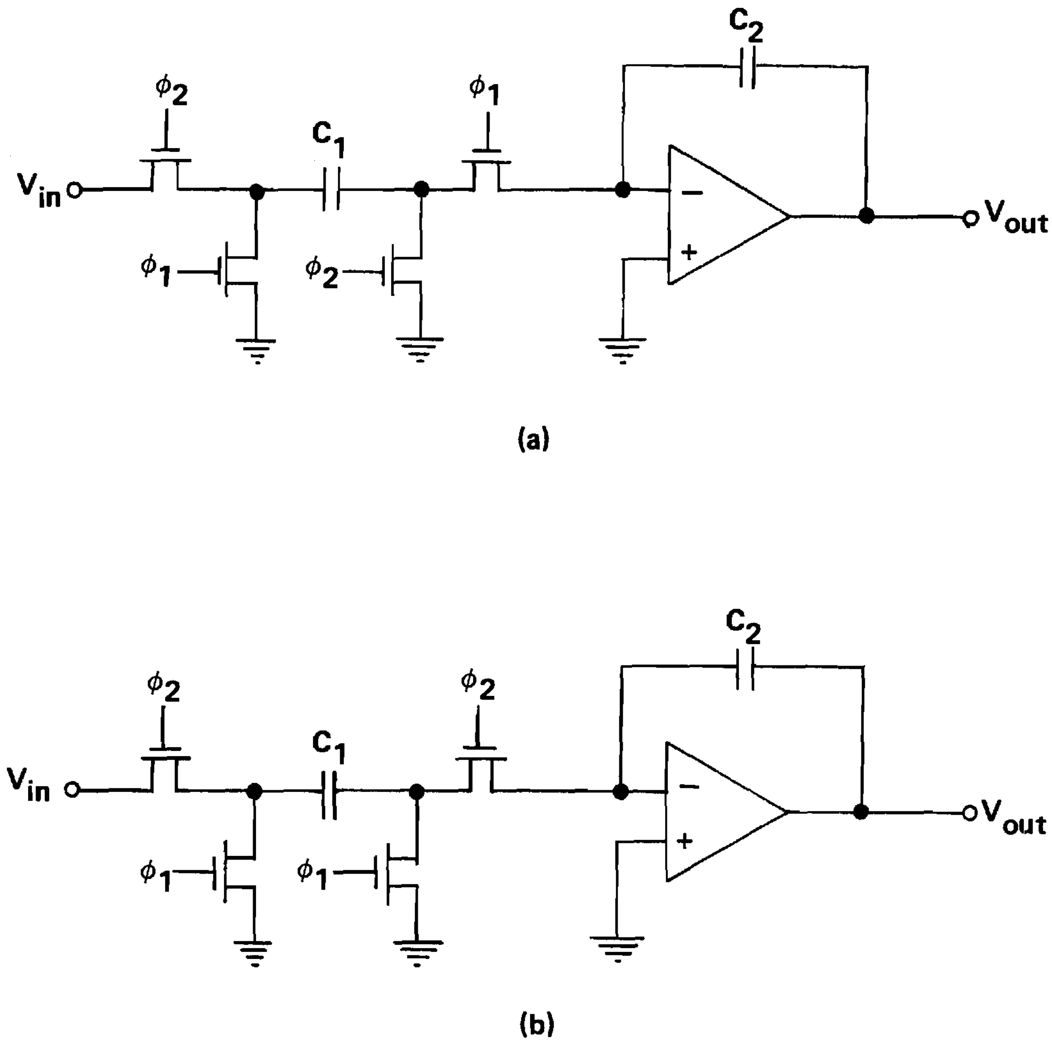


Fig. 2.6 Complementary integrators which are insensitive to stray capacitances. (a) Noninverting and (b) Inverting.

and ground. This enables filter designs to employ very small capacitances and thus minimize the IC chip area.

The mathematical analysis of the noninverting integrator is identical to that previously given in Eqs. 2-5 through 2-12. However, the reversed clock phasing sequence interchanges the plates of C_1 for transfering charge C_2. Thus, the negative sign in Eq. 2-8 becomes positive, or

$$C_2\, V_{out}(nT_C) = C_2\, V_{out}\left[(n-1)T_C\right] + C_1\, V_{in}\left[(n-1)T_C\right] \qquad (2\text{-}16)$$

Here we can utilize a short-cut by taking the z-transformation of Eq. 2-16 and recalling that $z=e^{sT}$.

$$C_2\, V_{out} = C_2\, V_{out} Z^{-1} + C_1\, V_{in} Z^{-1} \qquad (2\text{-}17)$$

Rearranging we have the desired transfer function in the z-variable, or

$$\frac{V_{out}}{V_{in}} = \frac{C_1/C_2}{Z - 1} \qquad (2\text{-}18)$$

Note, the sign is positive indicating the noninverting operation. Eq. 2-18 approximates the ideal integrator for $\omega T_C << 1$, or

$$\frac{V_{out}}{V_{in}} = \frac{C_1/C_2}{\cos \omega T_C + j \sin \omega T_C - 1} \approx \frac{C_1/C_2}{j\omega T_C} \qquad (2\text{-}19)$$

The analysis of the parasitic-free inverting integrator of Fig. 2.6(b) is identical to that given above with only a change of sign necessary. However, this circuit does have one important limitation. During the ϕ_2 clock phase, the input is coupled directly to the output through both capacitors, and any "instantaneous" changes in V_{in} will be felt at the output. Consequently, V_{in} must be held constant during the ϕ_2 phase. This can be done by preceeding this stage with a sample-and-hold circuit, and sampling the output voltage during only the ϕ_1 phase.

III. LOW PASS FILTER

A. Resonator Analog Active Filter

Figure 3.1 shows the basic analog active filter selected for low-pass and bandpass implementations. This circuit is described extensively in the literature and will not be repeated here [7,8]. This configuration, called a resonator or state-variable active filter offers these advantages:

1. Has reduced sensitivity to op amp gain bandwidth product;
2. Has grounded noninverting terminal of op amp for reduced sensitivity to input and output impedances;
3. Provides high Q; and
4. Permits separate and independent adjustments of gain, bandwidth (or Q), and center frequency or pole frequency.

Provisions for separate and independent adjustments of key parameters can be seen from the transfer functions

$$\text{Low-pass:} \quad \frac{V_{LP}}{V_{in}} = \frac{1/R_4C_2R_5C_1}{s^2 + s/R_3C_1 + 1/R_4C_2R_7C_1} \tag{3-1}$$

$$\text{Bandpass:} \quad \frac{V_{BP}}{V_{in}} = \frac{s/R_5C_1}{s^2 + s/R_3C_1 + 1/R_4C_2R_7C_1} \tag{3-2}$$

Equation (3-1) for the low-pass filter is of the general form

$$\frac{V_{LP}}{V_{in}} = \frac{G_o\omega_o^2}{s^2 + s(\omega_o/Q) + \omega_o^2} = \frac{G_o\omega_o^2}{s^2 + sB + \omega_o^2} \tag{3-3}$$

The dc gain is given by

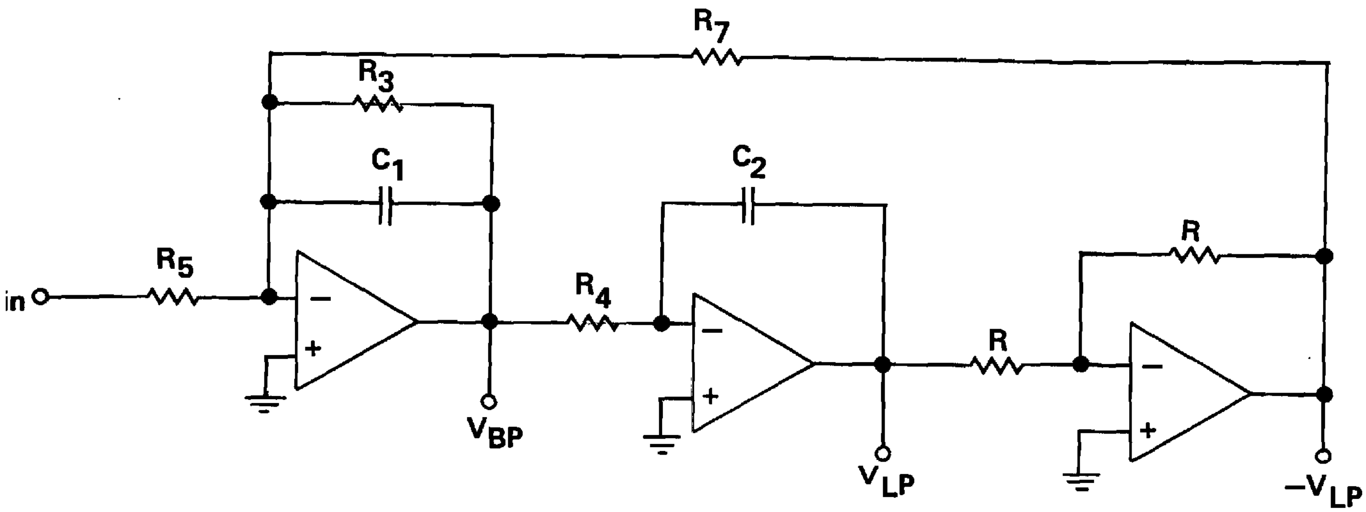


Fig. 3.1 **Resonator analog active filter for realization of second–order lowpass and bandpass functions.**

$$G_o = \frac{R_7}{R_5} \tag{3-4}$$

The pole frequency f_o is

$$f_o = \frac{\omega_o}{2\pi} = \frac{1}{2\pi\sqrt{R_4C_2\ R_7C_1}} \tag{3-5}$$

The filter bandwidth and Q are given by

$$B = 2\pi(BW) = 1/R_3C_1 \tag{3-6}$$

$$Q = \frac{\omega_o}{B} = R_3\sqrt{\frac{C_1}{R_4R_7C_2}} \tag{3-7}$$

Note, that R_5 will adjust the gain G_o without affecting f_o or Q. Also R_3 affects only the bandwidth or Q. The pole frequency depends on R_4, R_7, C_1, and C_2. When tuning, f_o should be adjusted first so as to not influence G_o or Q.

The resistors in the resonator circuit will be replaced with switched capacitors in the sampled-data implementation. For each subscripted resistor we can substitute

$$R_k = \frac{T_C}{\alpha_k\ C_{ref}} \quad \text{and} \quad \alpha_k = \frac{C_k}{C_{ref}} \tag{3-8}$$

The α's defined in this way then are the capacitance ratios between each switched capacitor and a non-switched, reference capacitor. The integrating capacitors (C_1 and C_2) will be taken as the reference elements. For example, α_3 is the ratio of the switched capacitor replacing R_3 to the integrating capacitor C_1. Odd values of α will be referenced to C_1. Even

α values are referenced to C_2. After Eq. (3-8) is substituted into Eq. (3-1), we have

$$\frac{V_{LP}}{V_{in}} = \frac{\alpha_4\alpha_5/T_C^{\ 2}}{s^2 + (\alpha_3/T_C)s + \alpha_4\alpha_7/T_C^{\ 2}} \tag{3-9}$$

Sedra and Brackett [9] suggest as a "rule of thumb" setting $\alpha_4 = \alpha_7$ to lessen potential problems with the dynamic range of the operational amplifiers in the circuit. However, to provide maximum flexibility, the designer can choose the desired ratio between α_4 and α_7 according to

$$\alpha_R = \alpha_7/\alpha_4 \tag{3-10}$$

With α_R specified, α_4 can be found from

$$\alpha_4 = \omega_o T_C/\sqrt{\alpha_R} = 2\pi\ f_o T_C/\sqrt{\alpha_R} \tag{3-11}$$

The filter Q from Eq. (3-7) reduces to

$$Q = \frac{\alpha_4\ \sqrt{\alpha_R}}{\alpha_3} \tag{3-12}$$

The dc gain simplifies to

$$G_o = \frac{\alpha_5}{\alpha_7} \tag{3-13}$$

In a low-pass filter design, f_o, Q, and G_o would be known parameters. A sampling clock frequency ($1/T_C$) would be chosen. The α values would then be calculated in the following order:

$$B = 2\pi\ f_o/Q \tag{3-14}$$

$$\alpha_3 = BT_C \tag{3-15}$$

$$\alpha_4 = 2\pi\ f_o T_C/\sqrt{\alpha_R} \tag{3-16}$$

$$\alpha_7 = \alpha_R \alpha_4 \tag{3-17}$$

$$\alpha_5 = \alpha_7 G_o \tag{3-18}$$

For the second-order Butterworth Filter

$$Q = \frac{1}{\sqrt{2}} = 0.7071 \tag{3-19}$$

Therefore, the only information the designer must specify is f_o, T_C, G_o, and α_R.

A low-pass Chebyshev Filter will have some peaking in its response as shown in Fig. 3.2. Here A_p is the ripple width in dB, f_p is the frequency where the magnitude of the response leaves the pass-band, and f_{refz} is the frequency of the reflection zero which corresponds to the frequency where the response peaks. (Note that neither f_p or f_{refz} will correspond to the pole frequency in the transfer function).

In a Chebyshev design, f_p, A_p, and G_o are specified. From this information, f_o, Q, and the bandwidth (in hertz) are calculated according to Sedra and Brackett [10] as follows:

$$\varepsilon = \sqrt{10^{A_p/10} - 1} \tag{3-20}$$

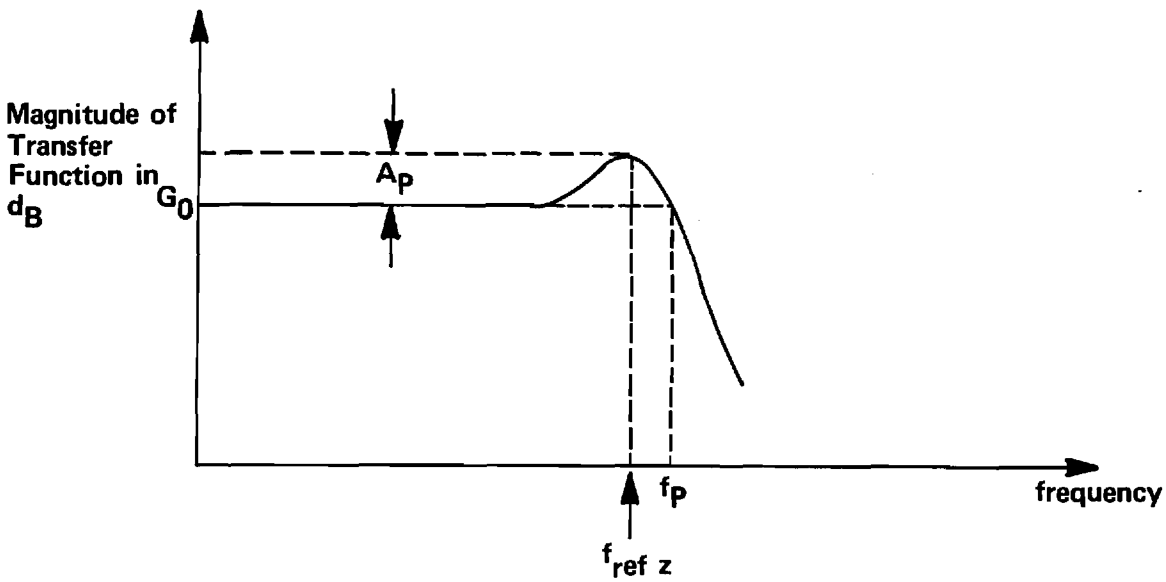


Fig. 3.2 Lowpass Chebyshev response.

$$\text{ARG1} = \frac{1}{2} \ln[1/\varepsilon + \sqrt{1/\varepsilon^2 + 1}] \tag{3-21}$$

$$Q = \frac{1}{2\cos\{\arctan[\cosh(\text{ARG1})/\sinh(\text{ARG1})]\}} \tag{3-22}$$

$$f_o = f_p\sqrt{\sinh^2(\text{ARG1}) + \frac{1}{2}} \tag{3-23}$$

$$BW = f_o/Q \tag{3-24}$$

A Generalized, second-order low-pass filter is completely specified by choosing f_o, Q (or B), G_o, α_R, and T_C. With this information, Eqs. 3-14 through 3-18 are used to determine all the α values required.

B. Z-Transformation

Refer to the analog resonator circuit of Fig. 3.1. This circuit can be transformed to the block diagram shown in Fig. 3.3(a) in the Z-variable. Note that each resistor is replaced by its corresponding α value. The inverting integrators are represented as rectangular blocks according to Eq. 2-10 and the $Z=e^{sT}$ identity. The transfer function for the low-pass and bandpass outputs are

$$\text{Low-pass:} \quad \frac{V_{LP}}{V_{in}} = \frac{\alpha_4\alpha_5}{Z^2 + (\alpha_3 - 2)Z + 1 - \alpha_3 + \alpha_4\alpha_7} \tag{3-25}$$

$$\text{Bandpass:} \quad \frac{V_{BP}}{V_{in}} = \frac{-\alpha_5(Z-1)}{Z^2 + (\alpha_3 - 2)Z + 1 - \alpha_3 + \alpha_4\alpha_7} \tag{3-26}$$

Note that both denominators are identical. Also note that as $\omega \to 0$, the transfer function for the low-pass filter approaches $G_o = \alpha_5/\alpha_7$.

The block diagram of Fig. 3.3(a) can be simplified when it is recognized that a noninverting integrator can be substituted for the last

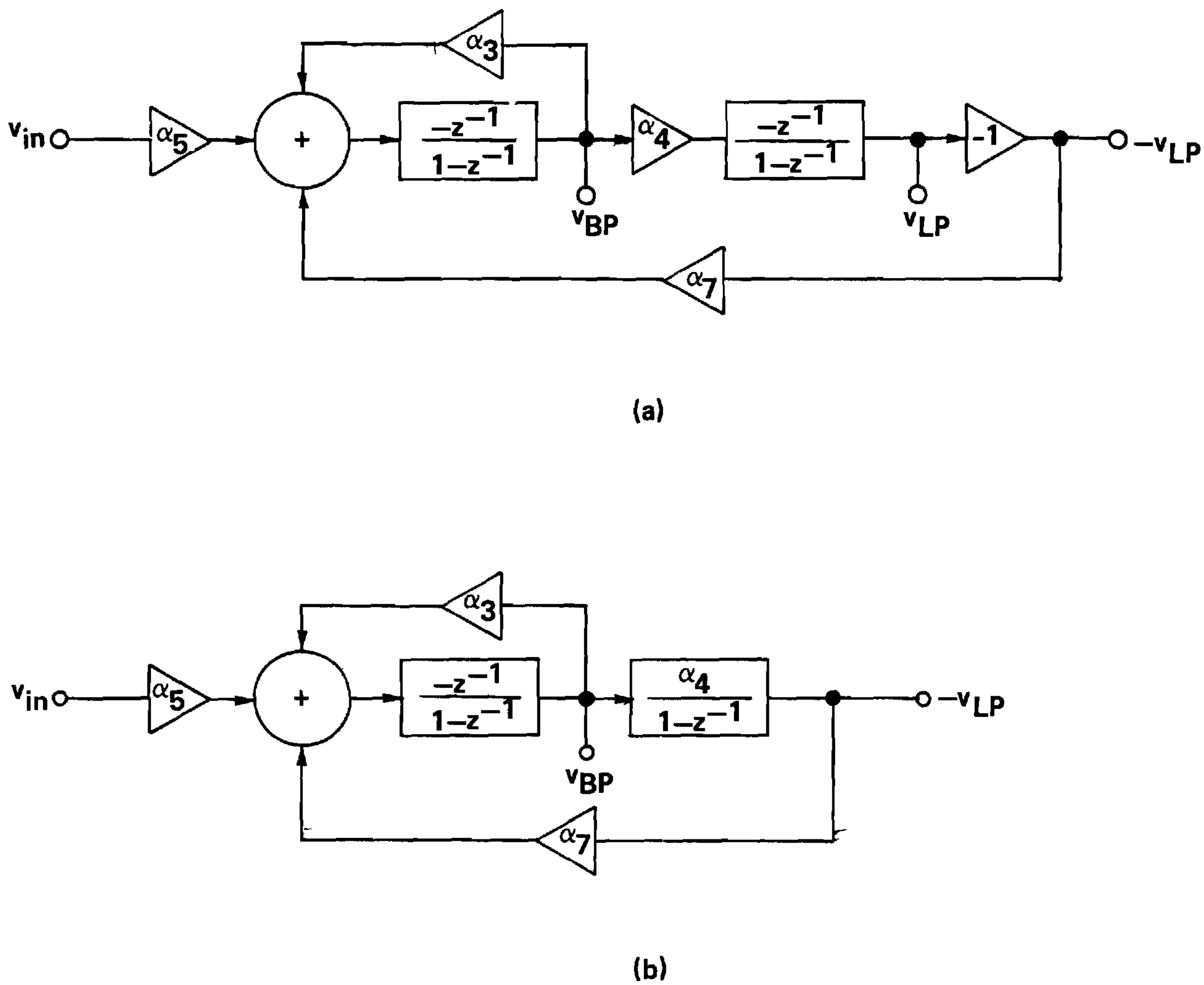


Fig. 3.3 Block diagram representation of the analog resonator in the z–variable (a) complete representation of Fig. 3.1 and (b) simplified version.

two stages. From Eq. 2-18 it should be noted that there is no Z^{-1} delay term in the numerator of the positive integrator. Hence, Fig. 3.3(b) shows a simpler version with the Z^{-1} change made. The low-pass and bandpass transfer functions for Fig. 3.3(b) are:

$$\text{Low-pass:} \quad \frac{V_{LP}}{V_{in}} = \frac{\alpha_4 \alpha_5 Z}{Z^2 + Z(\alpha_3 - 2 + \alpha_4 \alpha_7) + 1 - \alpha_3} \qquad (3.27)$$

$$\text{Bandpass:} \quad \frac{V_{BP}}{V_{in}} = \frac{-\alpha_5 (Z - 1)}{Z^2 + Z(\alpha_3 - 2 + \alpha_4 \alpha_7) + 1 - \alpha_3} \qquad (3\text{-}28)$$

Again note that both denominators are identical and that the low-pass function approaches G_o as $\omega \to 0$.

A SCF implementation of Fig. 3.3(b) is given in Fig. 3.4. Sampling at both outputs should be done only during the ϕ_1 clock phase. Also, it would be advisable to preceed the input with a sample-and-hold circuit. If the input signal were to change during ϕ_2, this change would be felt immediately at the bandpass output.

The magnitude and phase of the low-pass and bandpass functions can be determined at any arbitrary frequency, ω, by first making the following substitutions:

$$Z = \cos \omega T_C + j \sin \omega T_C \qquad (3\text{-}29)$$

$$Z^2 = \cos 2\omega T_C + j \sin 2\omega T_C \qquad (3\text{-}30)$$

The low-pass filter transfer function now becomes

$$\frac{V_{LP}}{V_{in}} = \frac{\alpha_4 \alpha_5 (\cos \omega T_C + j \sin \omega T_C)}{\cos 2\omega T_C + (\alpha_3 - 2 + \alpha_4 \alpha_7) \cos \omega T_C + 1 - \alpha_3 + j[\sin 2\omega T_C + (\alpha_3 - 2 + \alpha_4 \alpha_7) \sin \omega T_C]}$$

(3-31)

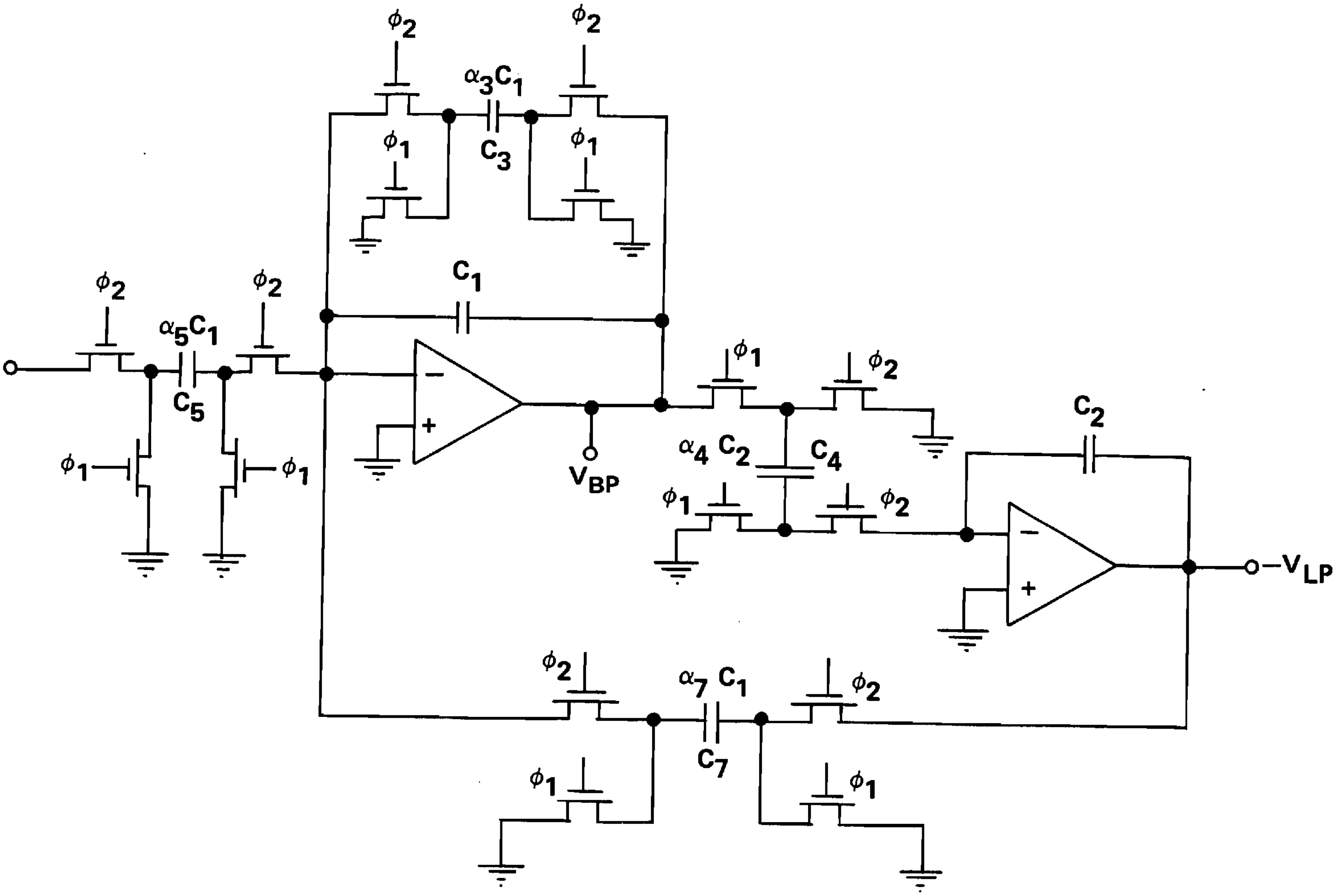


Fig. 3.4 SCF implementation of Fig. 3.3(b) for the low–pass and bandpass filters.

This complex function can be evaluated in the usual manner to find the magnitude and phase angle as a function of the ω frequency.

C. MARTLP Computer-Aided Design Program

MARTLP is a computer-aided design program developed from the preceeding sections and based upon the switched-capacitor filters of Martin [11]. The program is an interactive, prompting routine which allows the user to select either the Butterworth, Chebyshev, or Generalized responses for a second-order system. The program is called by the following commands:

OLD MARTLP

FORTRAN

RUN-10 MARTLP

(The number in "RUN-10" specifies a maximum execution time in CPU seconds; 10 was used arbitrarily.)

After logging on, the menu of commands will appear as shown below.

```
NEED MENU?  (0=NO, 1=YES)
=1
A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,
ENTER THE NUMBER OF THE APPROPRIATE COMMAND.

1.  ENTER PARAMETERS FOR BUTTERWORTH LOWPASS FILTER
2.  ENTER PARAMETERS FOR CHEBYSHEV LOWPASS FILTER
3.  ENTER PARAMETERS FOR GENERALIZED LOWPASS FILTER
4.  PRINT MENU
5.  TERMINATE PROGRAM
6.  WARP FO, BW, AND GAIN.
7.  PRINT CURRENT VALUES.
8.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F
9.  FIND CAPACITOR VALUES AND TOTAL C
10.  CHANGE FO ONLY
11.  CHANGE CHEBYSHEV RIPPLE WIDTH ONLY
12.  CHANGE GENERALIZED FILTER BANDWIDTH ONLY
13.  CHANGE FC ONLY
14.  CHANGE ALPHAC=C(2)/C(1) ONLY
15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4) ONLY
16.  CHANGE GAIN AT F=0 ONLY (DECIBELS)
17.  CHANGE CHEBYSHEV PASSBAND EDGE, FP, ONLY.
18.  FIND MAGNITUDE AND PHASE AT A SERIES
     OF FREQUENCY POINTS.
19.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON FO & BW
20.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER
```

The designer then specifies either the Butterworth, Chebyshev, or Generalized filter type. He is then prompted to input the design parameters of f_o, G_o, T_C, etc. as appropriate for the type filter chosen. At this point the program will have enough information to calculate all α values and to calculate the frequency response. If a frequency response calculation is tried at this point, magnitude and phase errors will be apparent because of the non-zero sampling time, T_C. This effect, known as warping, must be corrected. Menu Command 6 corrects for frequency warping by changing f_o, BW, and gain values until the actual Z-transformed frequency response matches the desired response to within some prescribed accuracy.*

MARTLP warps f_o, BW, and G_o simultaneously. As an example, suppose the desired parameters are f_o = 1.0 kHz, BW = 100 Hz, and G_o = 20 dB. Suppose the calculated α values produce a frequency response where f_o = 1050 Hz, BW = 90 Hz, and the G_o = 19 dB. The warp algorithm then adjusts a new f_o' to 950 Hz, a new BW' to 110 Hz, and a new G_o' to 21 dB. Frequency response calculations are repeated with this new data, and new f_o, BW, and G_o values are determined. Comparisons are then made with the original f_o, BW, and G_o design parameters. If an error greater than 10 ppm exists between any new parameter and the corresponding design parameter, the WARP routine is automatically repeated until the accuracy criteria are realized. Now, the new α values compensate for finite frequency sampling.

Menu Command 9 is used to find capacitor values and the total circuit capacitance. To execute this command, the program will ask for one

---

*The programmed accuracy in the MARTLP program is 10 ppm. An internal modification to the program is necessary if the user desires another accuracy limit.

or more "free" capacitor ratios to be specified, such as

$$\alpha_C = C_2/C_1 \tag{3-32}$$

Any real positive number can be inserted here. However, setting all free α's equal to unity is the simplest data to provide on the initial trial design. The proram will then ask for the minimum capacitor value planned for the actual SCF implementation. A value of C_{min} = 1 pF is a reasonable initial value. Finally, the capacitance density is required. A value of 0.2 pF/mil^2 is reasonable.

Menu Command 20 should be used in conjunction with Command 9. This routine permits the designer to alter any one of the "free" capacitor ratios over a prescribed range of values in incremental steps. In this algorithm, the program steps through the specified range of "free" capacitor ratios. The total circuit capacitance, C_{TOTAL}, is monitored. All α values which produce the minimum C_{TOTAL} are stored. If the minimum C_{TOTAL} occurs at the end point of a range, the designer should extend the range until the minimum falls inside the range. Minimizing the total circuit capacitance in this way also minimizes the spread between the largest and smallest capacitor values required in the circuit.

Experience with the MARTLP program indicates that the number of WARP iterations will be significantly reduced if the capacitance values are calculated first after data entry (Command 9). Next use Command 20 to minimize the total capacitance. Lastly, use Command 6 to warp these values to produce the desired response.

Command 19 permits the designer to sweep the clock frequency and observe the effects on f_o and BW. This command can also be used to evaluate the sensitivities of f_o and BW to clock frequency.

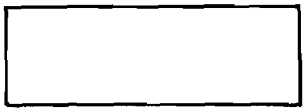
Commands 10 through 17 permit the user to alter one design parameter without re-entering all of the data. Following the making of such a change, the user then can proceed with any of the other menu commands he desires.

D. Flow Diagram for MARTLP

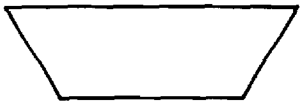
The flow chart showing the order and branching decisions for the low-pass SCF follows. The complete FORTRAN listing for MARTLP is given in Appendix A.

Flow Chart Symbols

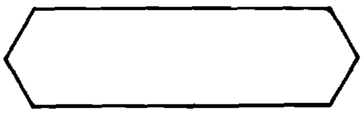
Beginning or ending



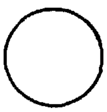
Arithmetic processing



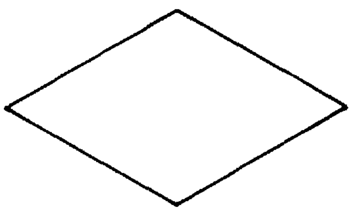
Input/output operation



Subroutine — subroutine name is given in parentheses inside the symbol



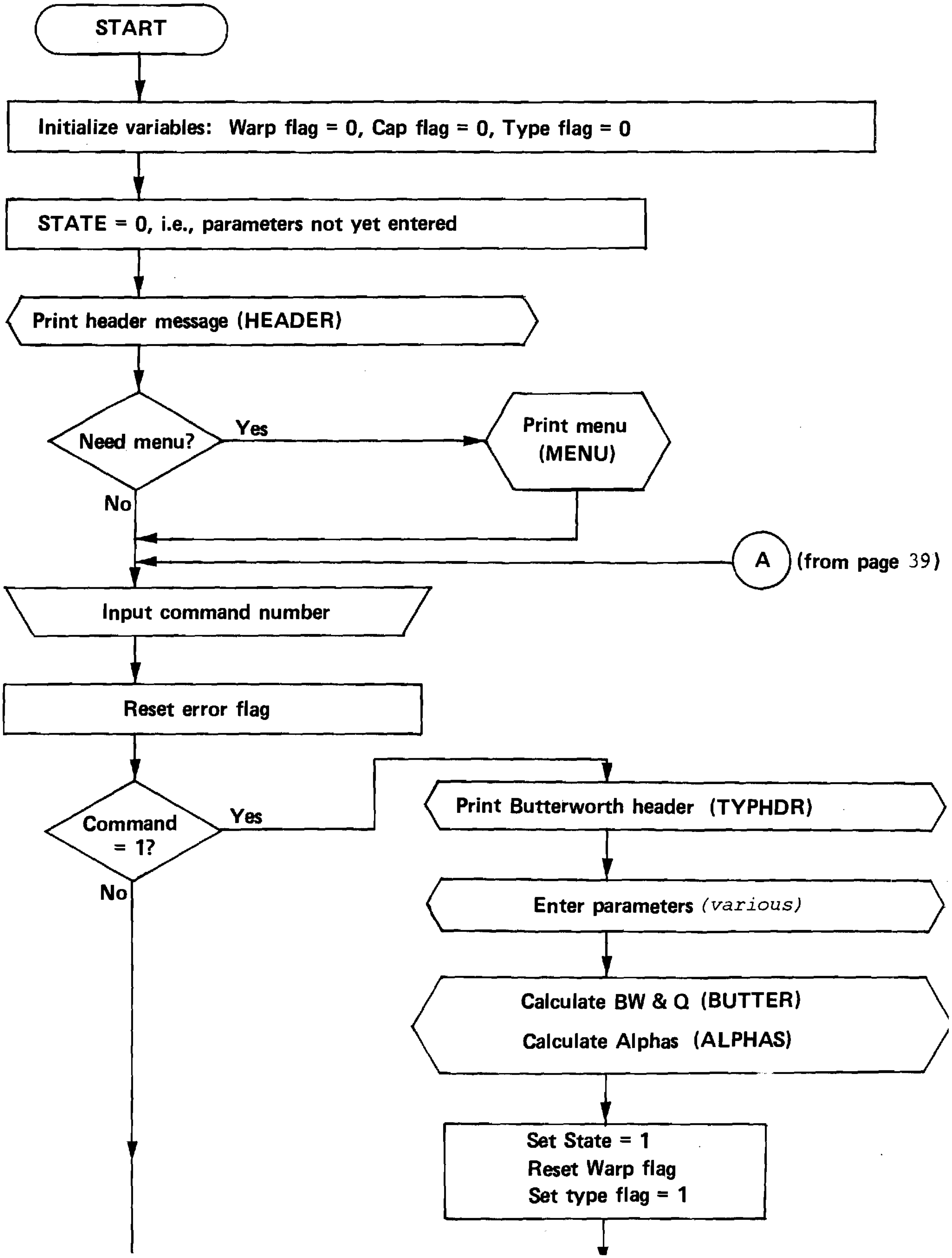
Connecting point

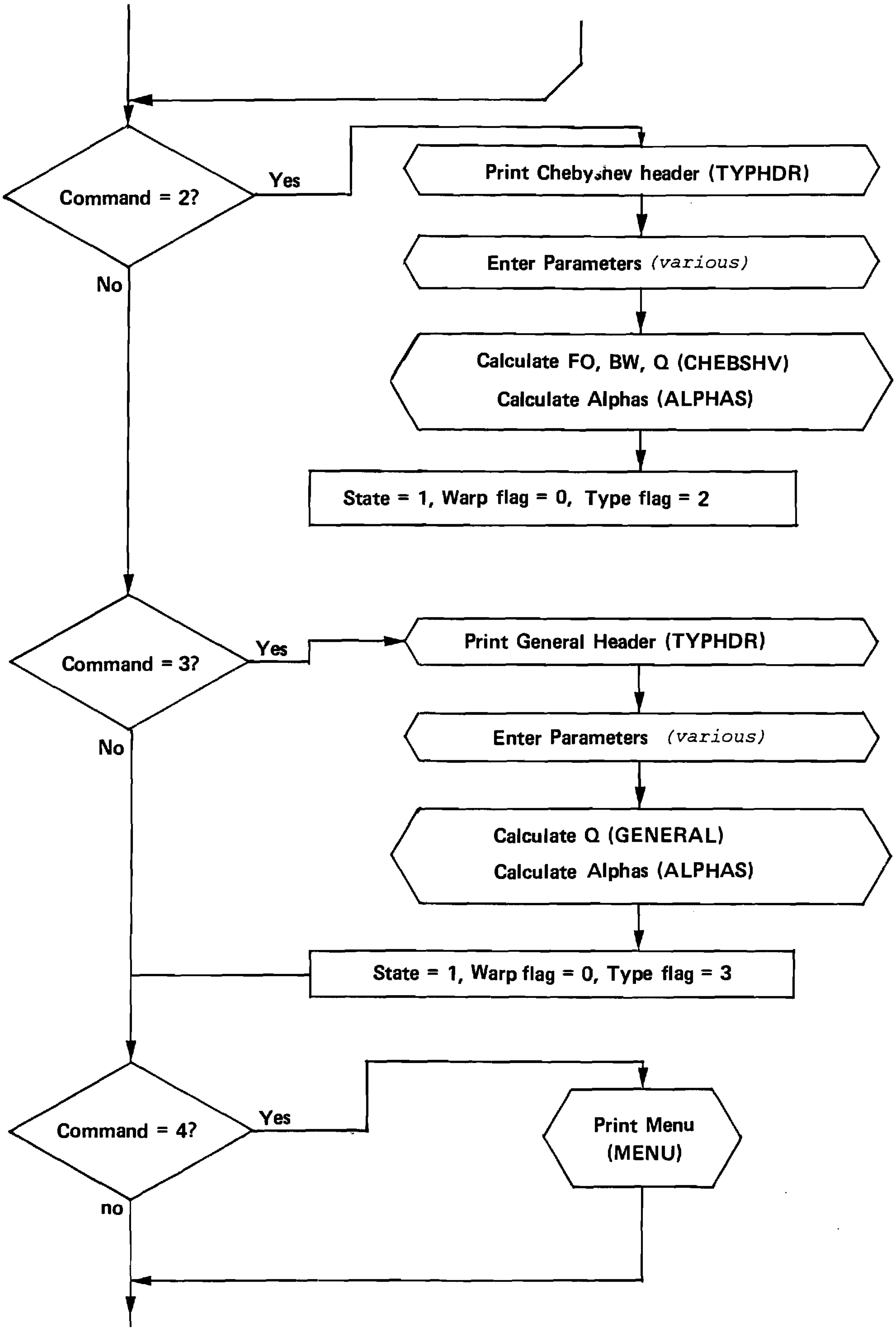


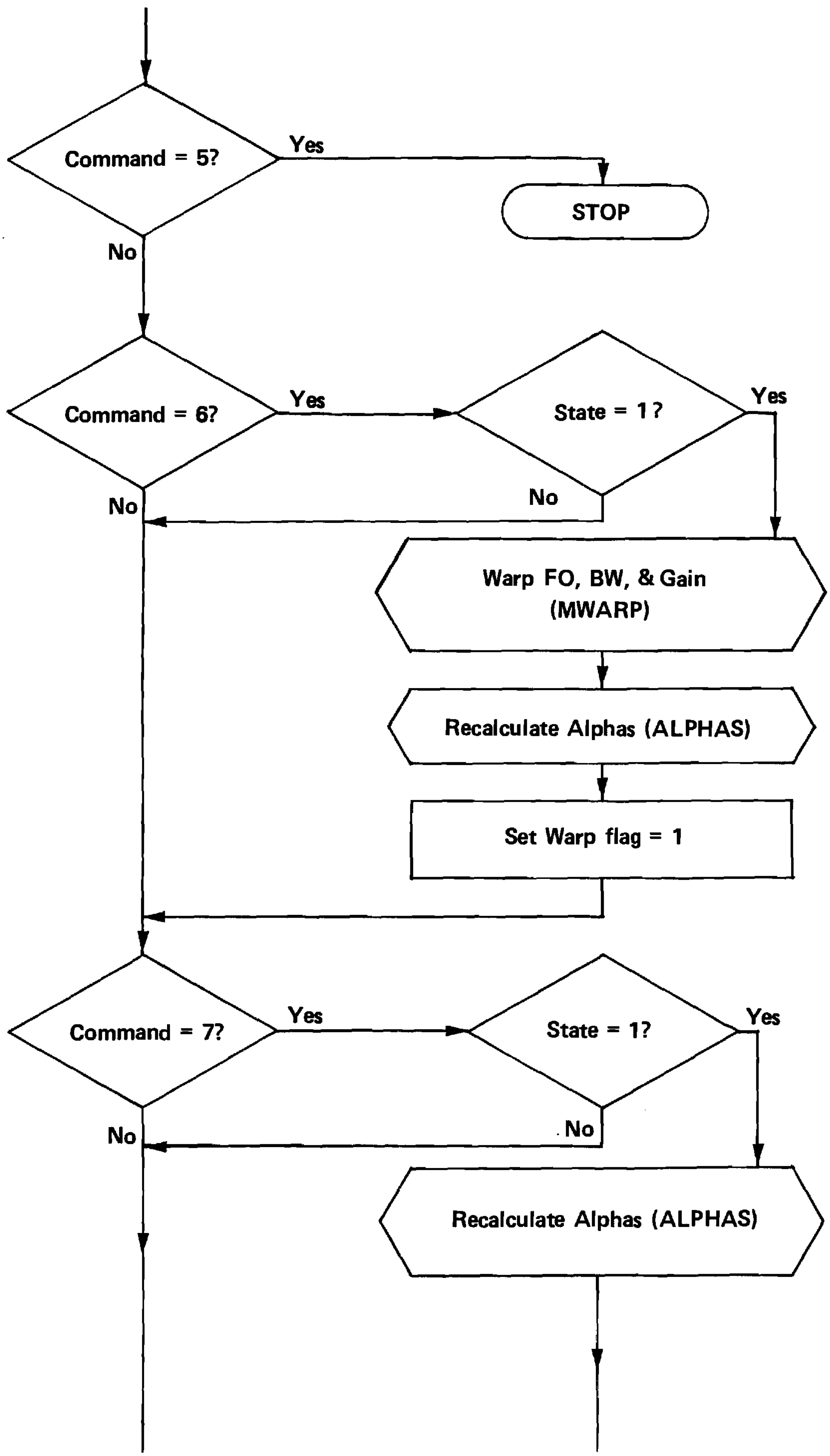
Decision Point

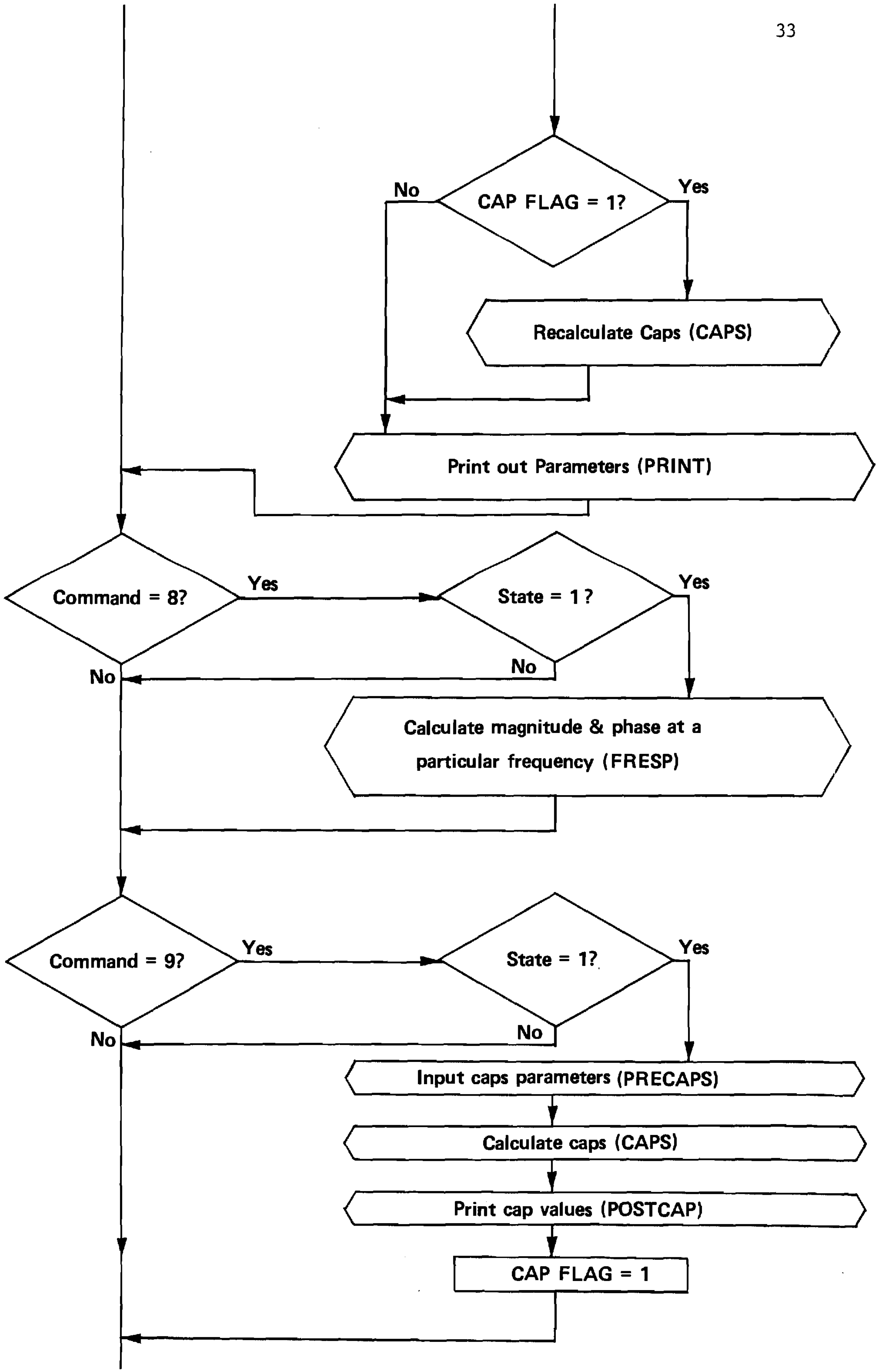
MARTLP Flowchart

Main Program

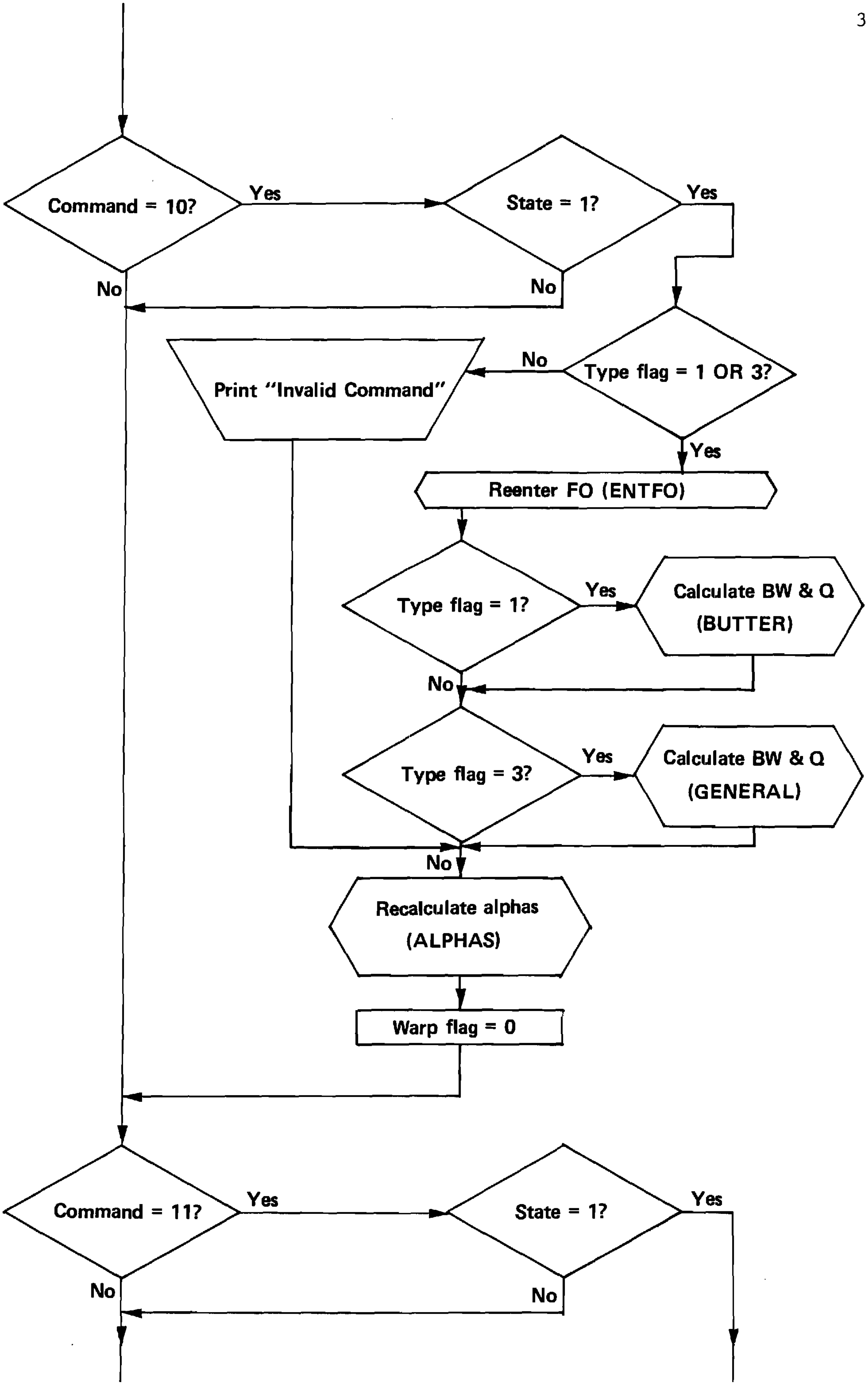



Command = 2?
Yes
No
Print Chebyshev header (TYPHDR)
Enter Parameters (various)
Calculate FO, BW, Q (CHEBSHV)
Calculate Alphas (ALPHAS)
State = 1, Warp flag = 0, Type flag = 2
Command = 3?
Yes
No
Print General Header (TYPHDR)
Enter Parameters (various)
Calculate Q (GENERAL)
Calculate Alphas (ALPHAS)
State = 1, Warp flag = 0, Type flag = 3
Command = 4?
Yes
no
Print Menu
(MENU)

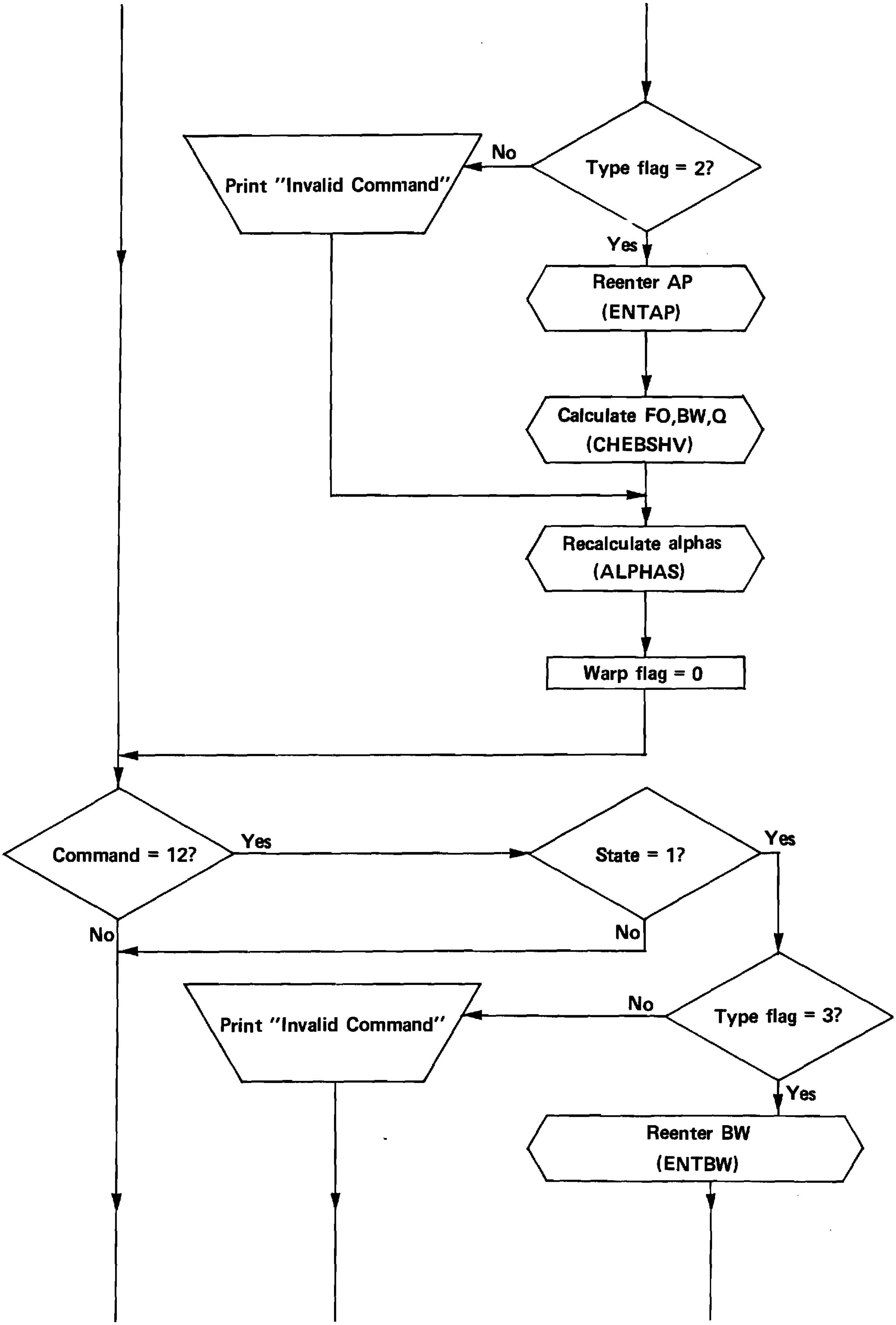

Command = 5?
Yes
STOP
No
Command = 6?
Yes
State = 1 ?
Yes
No
No
Warp FO, BW, & Gain (MWARP)
Recalculate Alphas (ALPHAS)
Set Warp flag = 1
Command = 7?
Yes
State = 1?
Yes
No
No
Recalculate Alphas (ALPHAS)



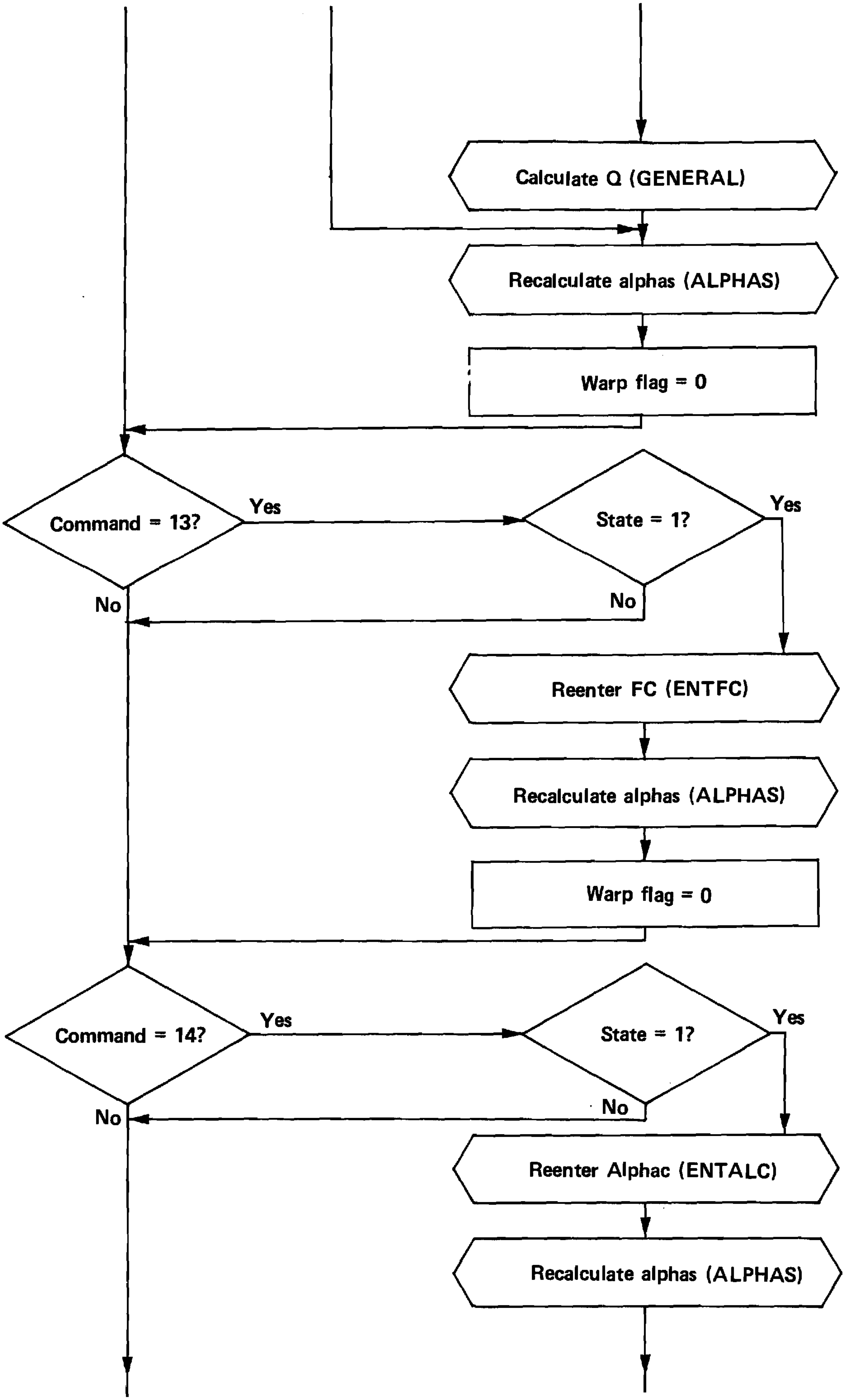
CAP FLAG = 1?
No
Yes
Recalculate Caps (CAPS)
Print out Parameters (PRINT)
Command = 8?
Yes
State = 1 ?
Yes
No
No
Calculate magnitude & phase at a particular frequency (FRESP)
Command = 9?
Yes
State = 1?
Yes
No
No
Input caps parameters (PRECAPS)
Calculate caps (CAPS)
Print cap values (POSTCAP)
CAP FLAG = 1

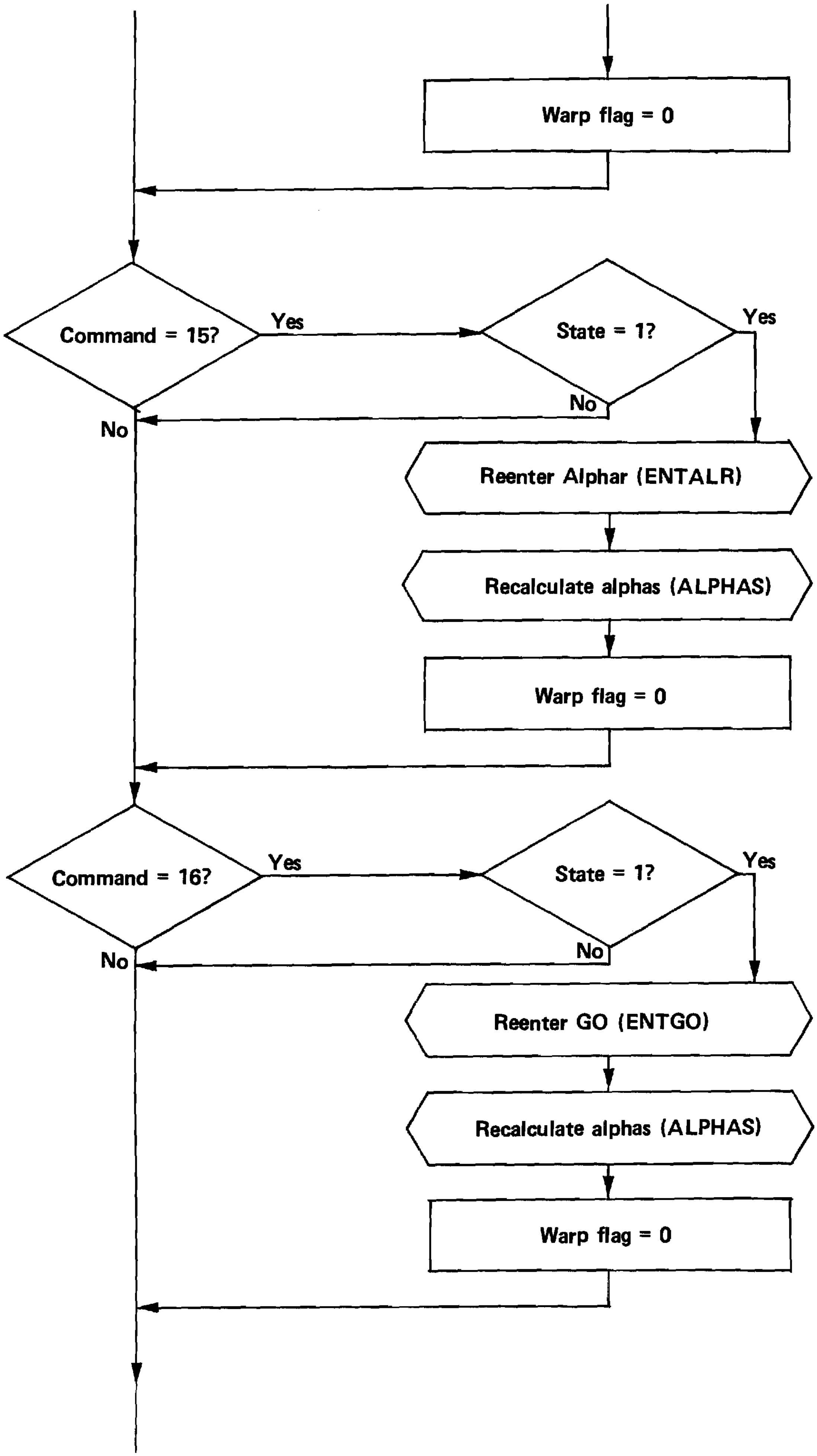


Command = 10?
Yes
State = 1?
Yes
No
No
Print "Invalid Command"
No
Type flag = 1 OR 3?
Yes
Reenter FO (ENTFO)
Type flag = 1?
Yes
Calculate BW & Q (BUTTER)
No
Type flag = 3?
Yes
Calculate BW & Q (GENERAL)
No
Recalculate alphas (ALPHAS)
Warp flag = 0
Command = 11?
Yes
State = 1?
Yes
No
No

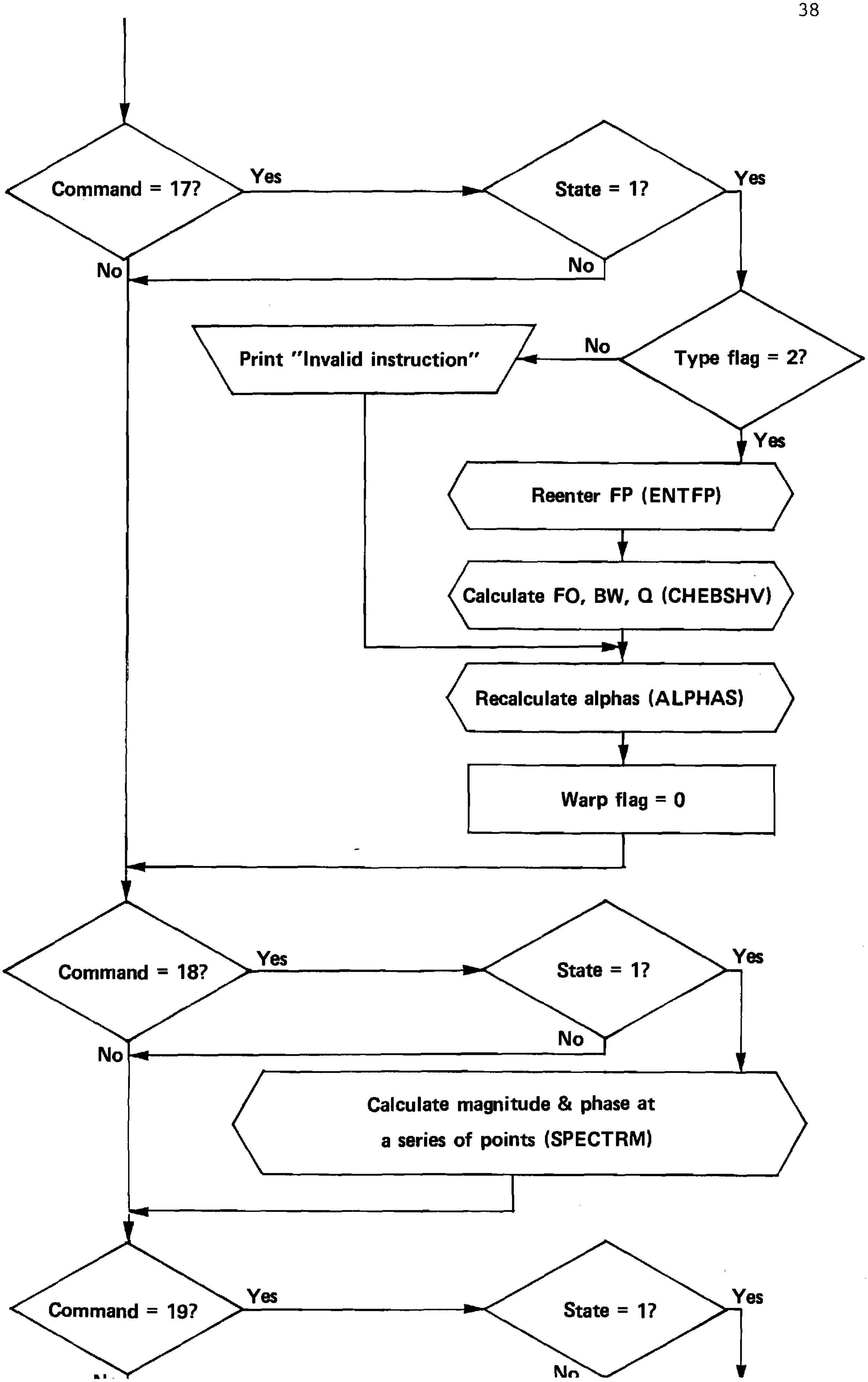


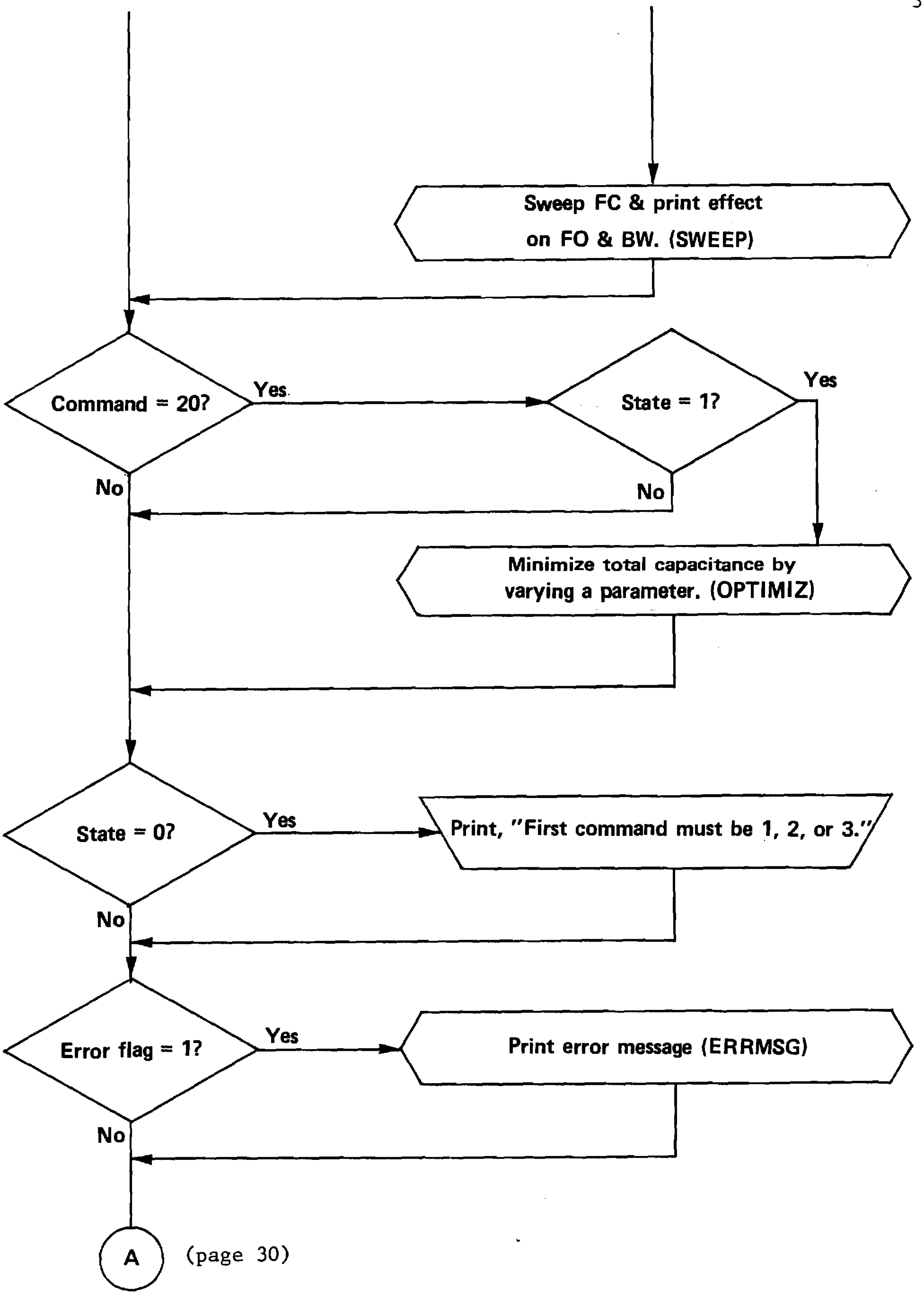
Type flag = 2?
No
Print "Invalid Command"
Yes
Reenter AP
(ENTAP)
Calculate FO,BW,Q
(CHEBSHV)
Recalculate alphas
(ALPHAS)
Warp flag = 0
Command = 12?
Yes
State = 1?
Yes
No
No
Type flag = 3?
No
Print "Invalid Command"
Yes
Reenter BW
(ENTBW)


Calculate Q (GENERAL)
Recalculate alphas (ALPHAS)
Warp flag = 0
Command = 13?
Yes
No
State = 1?
Yes
No
Reenter FC (ENTFC)
Recalculate alphas (ALPHAS)
Warp flag = 0
Command = 14?
Yes
No
State = 1?
Yes
No
Reenter Alphac (ENTALC)
Recalculate alphas (ALPHAS)

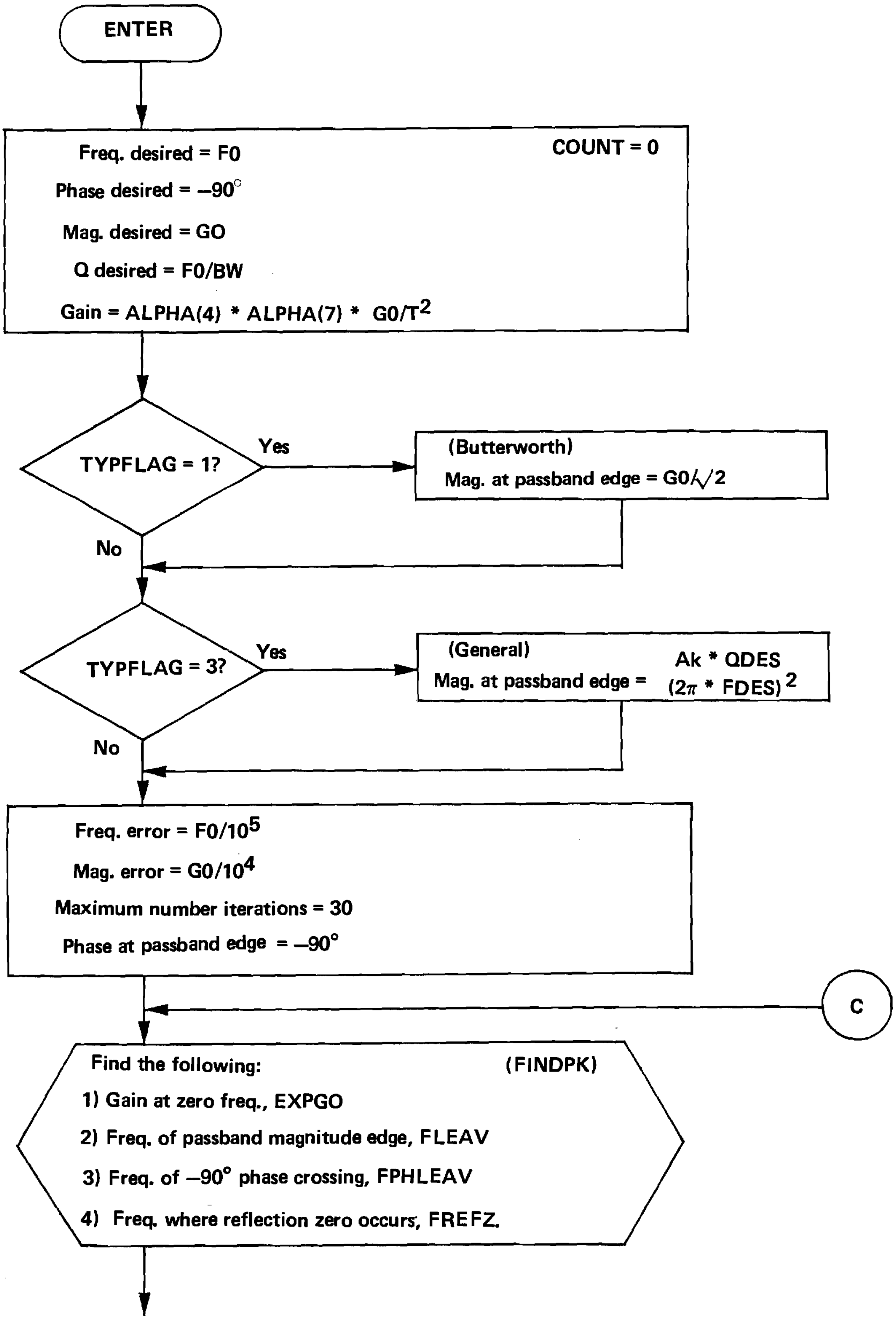


Warp flag = 0
Command = 15?
Yes
State = 1?
Yes
No
No
Reenter Alphar (ENTALR)
Recalculate alphas (ALPHAS)
Warp flag = 0
Command = 16?
Yes
State = 1?
Yes
No
No
Reenter GO (ENTGO)
Recalculate alphas (ALPHAS)
Warp flag = 0

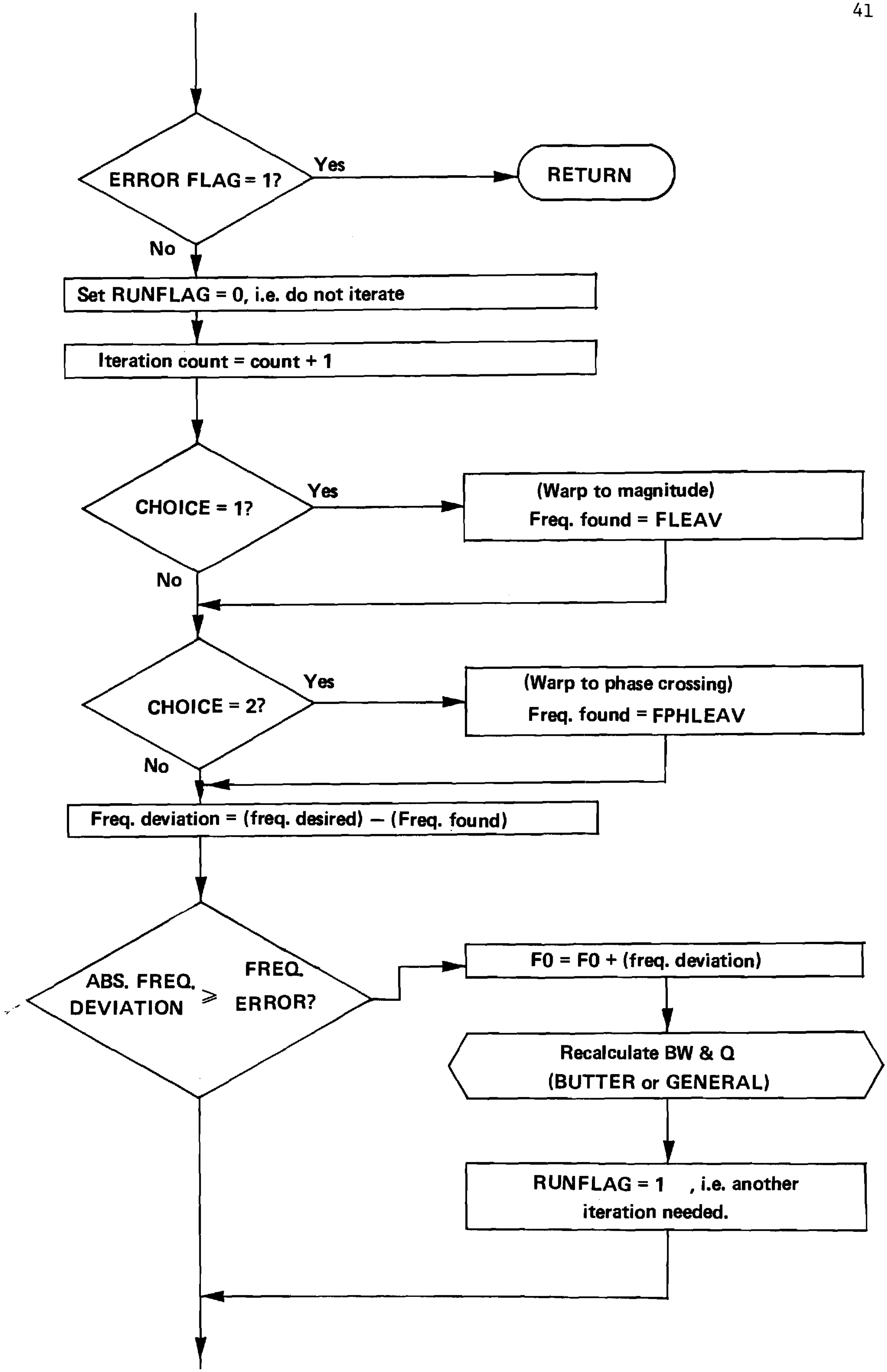

Command = 17?
Yes
State = 1?
Yes
No
No
Print "Invalid instruction"
No
Type flag = 2?
Yes
Reenter FP (ENTFP)
Calculate FO, BW, Q (CHEBSHV)
Recalculate alphas (ALPHAS)
Warp flag = 0
Command = 18?
Yes
State = 1?
Yes
No
No
Calculate magnitude & phase at
a series of points (SPECTRM)
Command = 19?
Yes
State = 1?
Yes
No

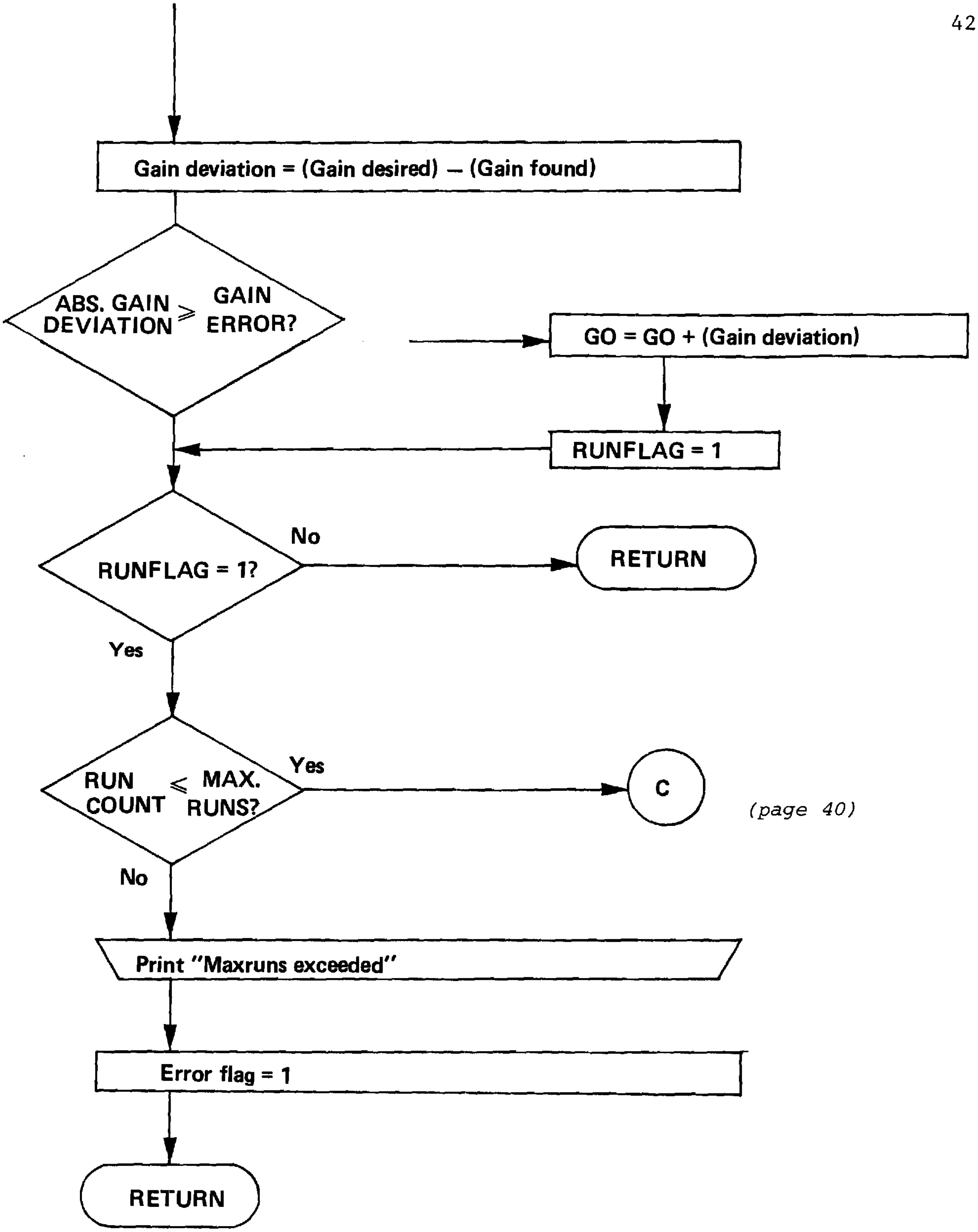

Sweep FC & print effect on FO & BW. (SWEEP)
Command = 20?
Yes
State = 1?
Yes
No
No
Minimize total capacitance by varying a parameter. (OPTIMIZ)
State = 0?
Yes
Print, "First command must be 1, 2, or 3."
No
Error flag = 1?
Yes
Print error message (ERRMSG)
No
A
(page 30)

Subroutine Warpbg

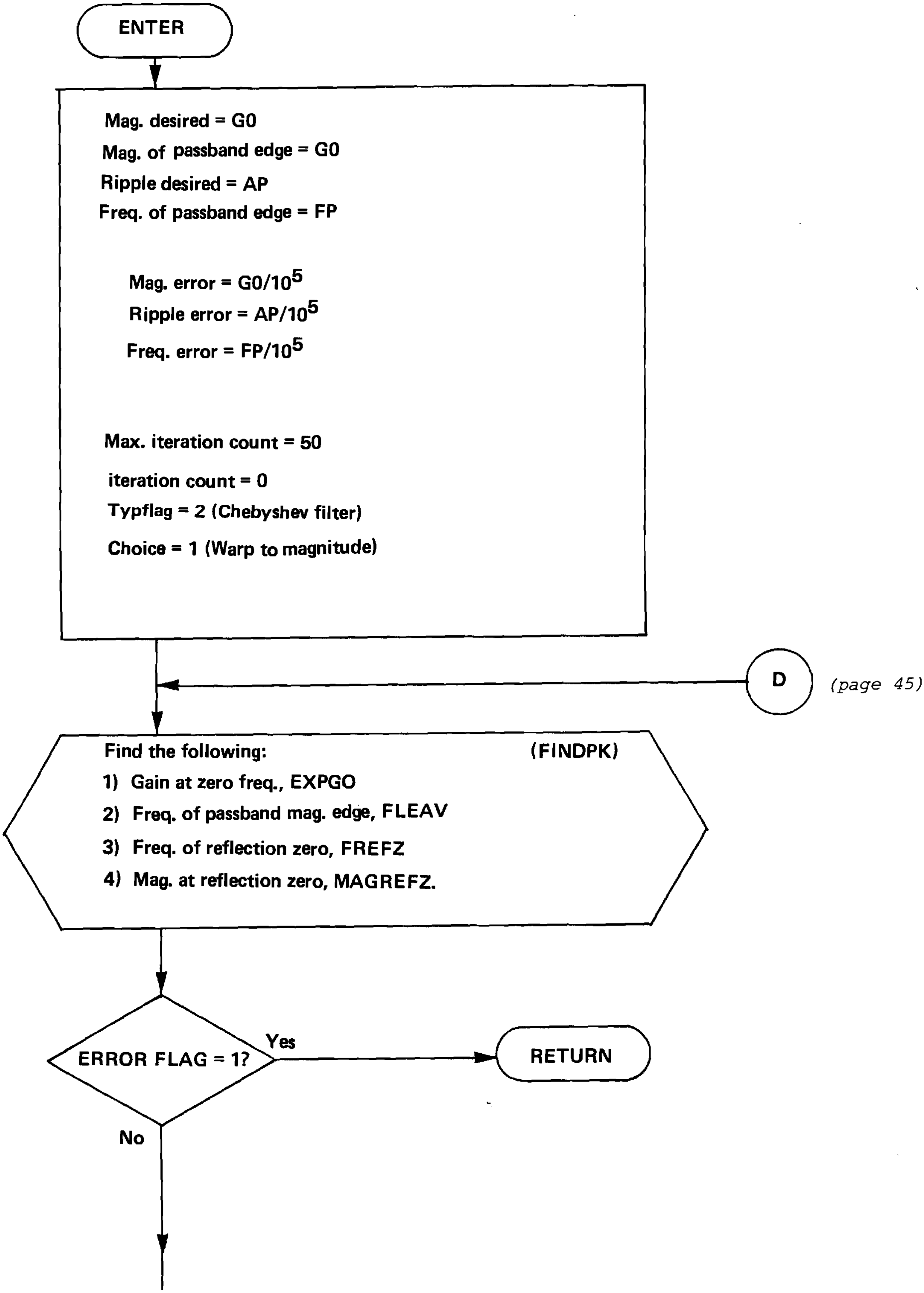


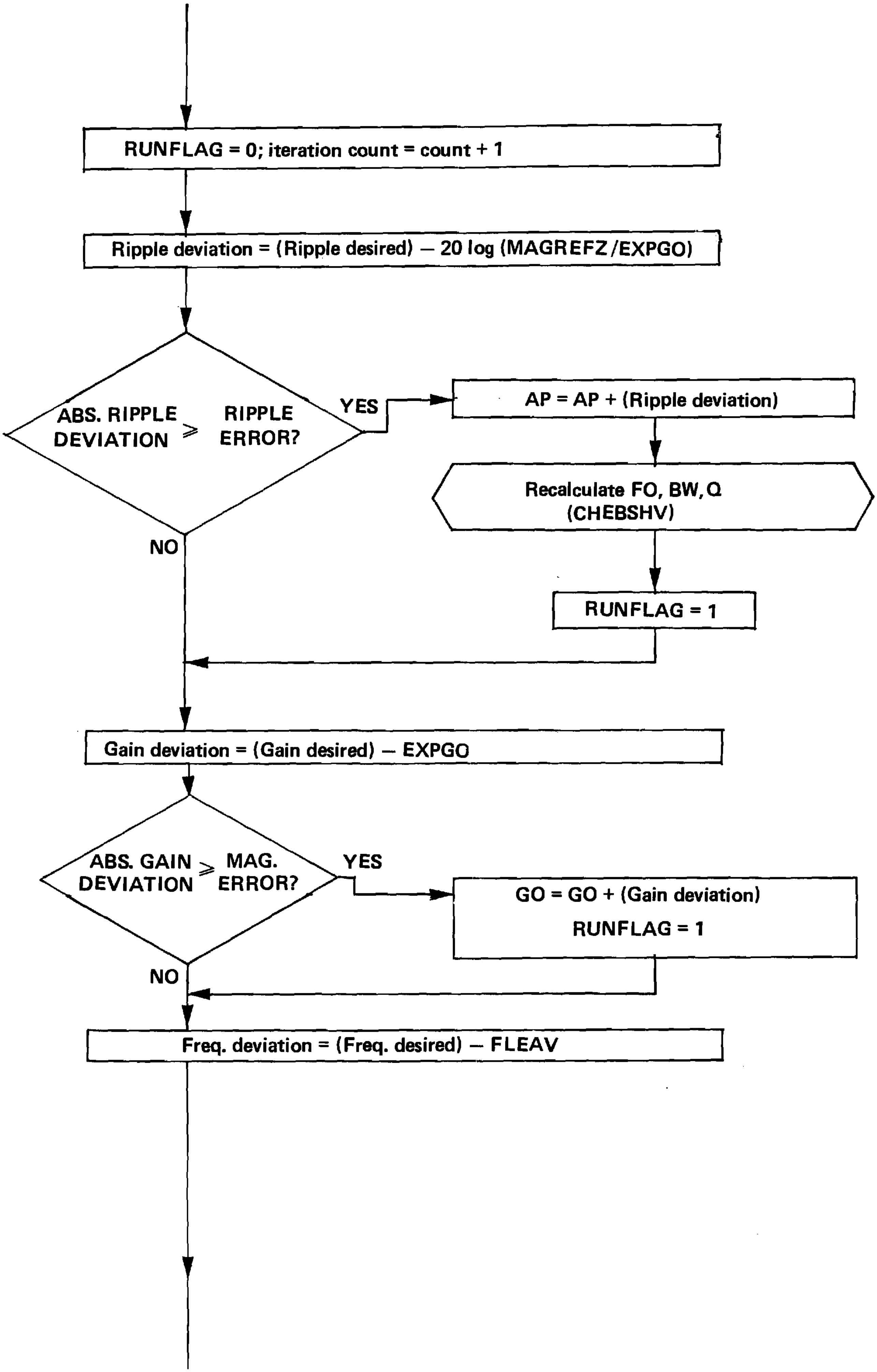
(page 42)

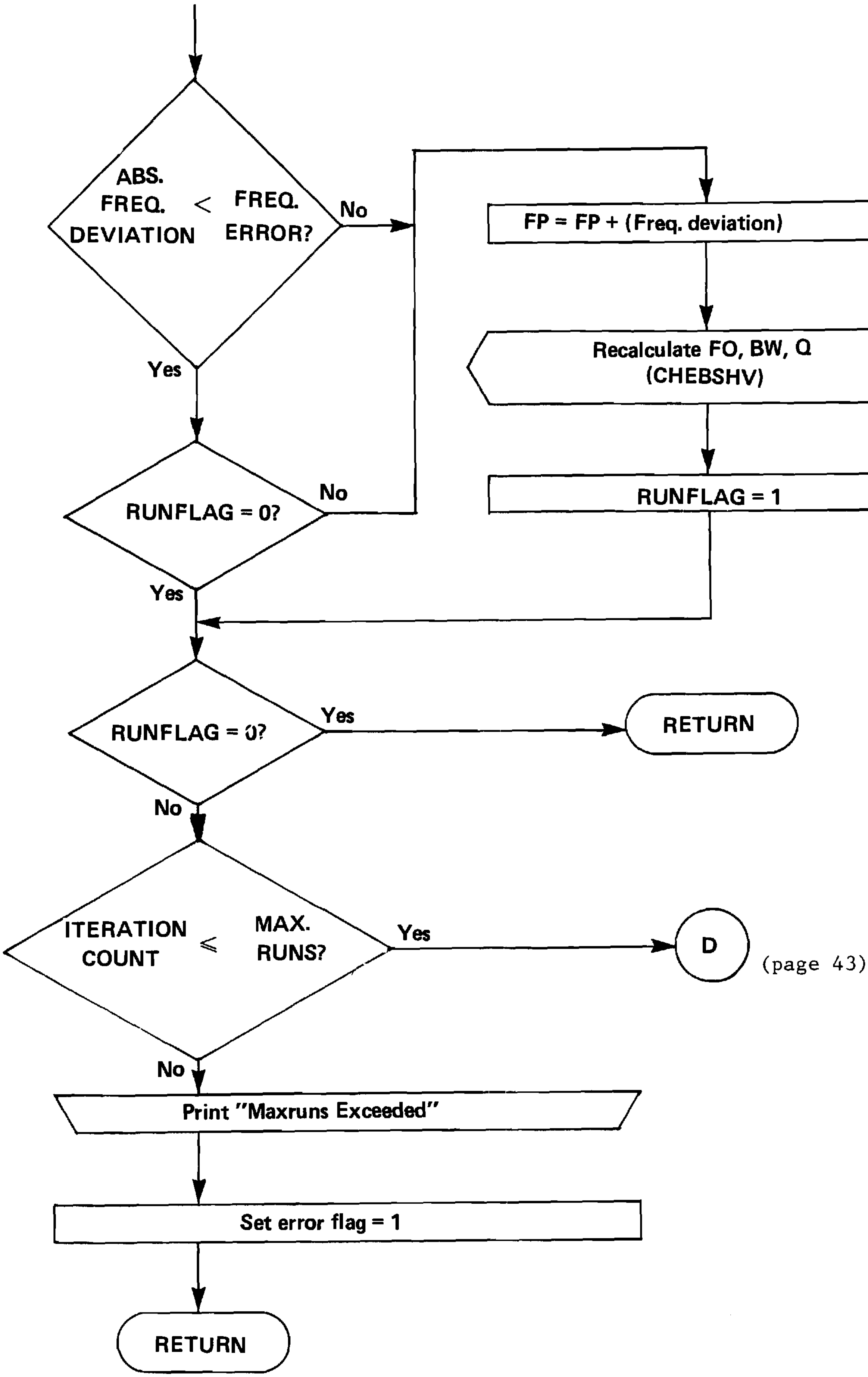

ERROR FLAG = 1?
Yes
RETURN
No
Set RUNFLAG = 0, i.e. do not iterate
Iteration count = count + 1
CHOICE = 1?
Yes
(Warp to magnitude)
Freq. found = FLEAV
No
CHOICE = 2?
Yes
(Warp to phase crossing)
Freq. found = FPHLEAV
No
Freq. deviation = (freq. desired) — (Freq. found)
ABS. FREQ. DEVIATION ⩾ FREQ. ERROR?
F0 = F0 + (freq. deviation)
Recalculate BW & Q
(BUTTER or GENERAL)
RUNFLAG = 1 , i.e. another
iteration needed.


Gain deviation = (Gain desired) — (Gain found)
ABS. GAIN DEVIATION ≥ GAIN ERROR?
GO = GO + (Gain deviation)
RUNFLAG = 1
RUNFLAG = 1?
No
RETURN
Yes
RUN COUNT ≤ MAX. RUNS?
Yes
C
No
Print "Maxruns exceeded"
Error flag = 1
RETURN

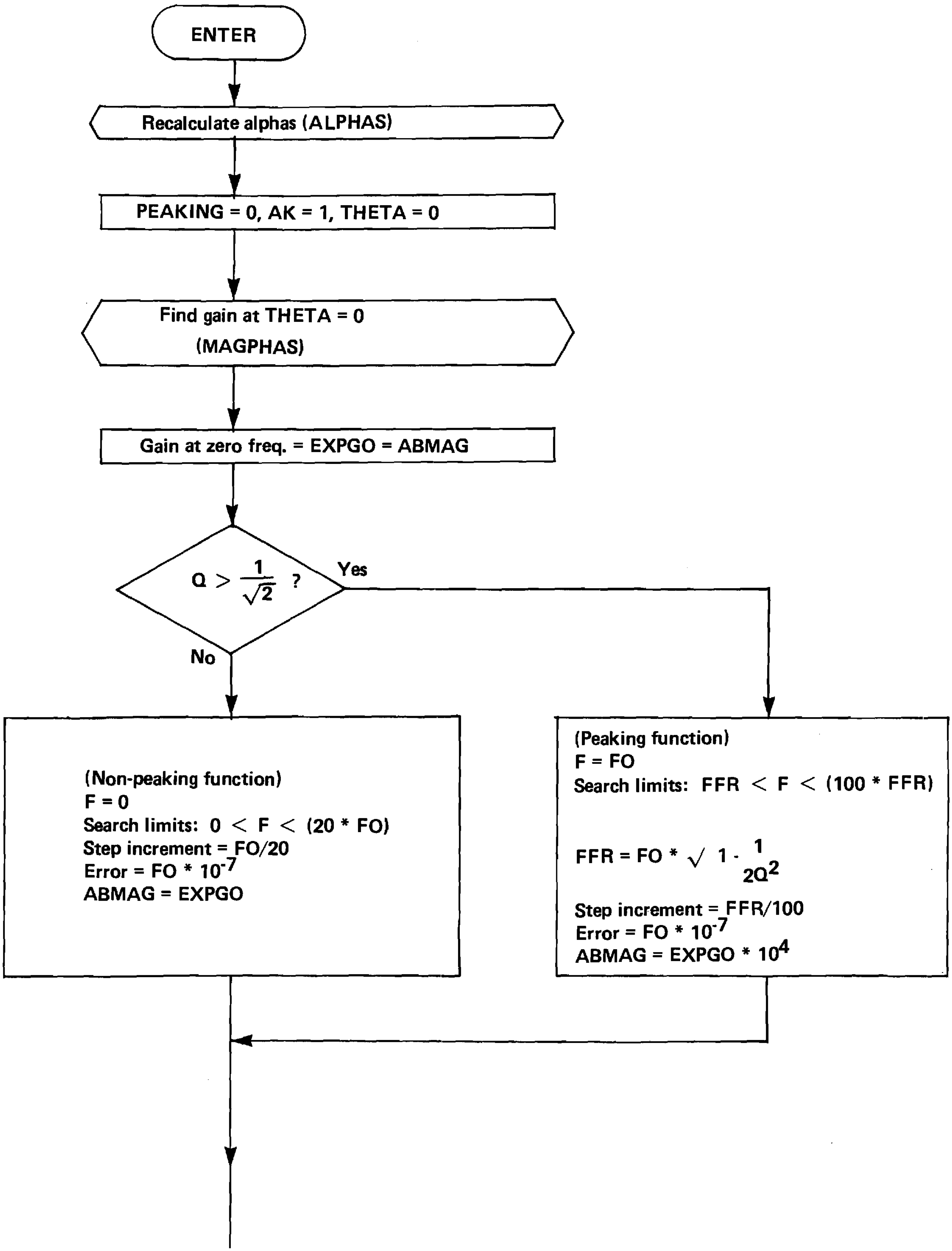
(page 40)

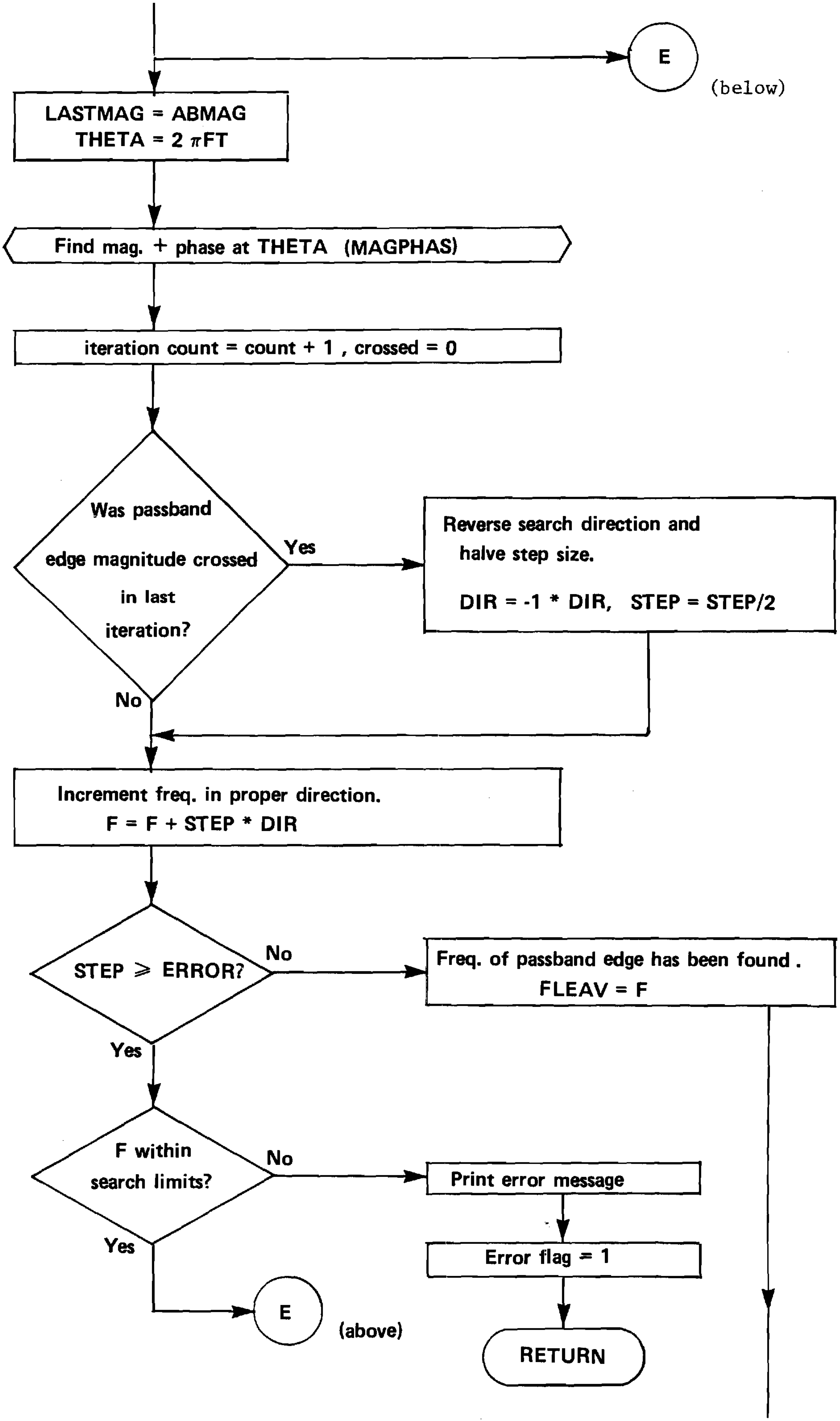

ENTER
Mag. desired = G0
Mag. of passband edge = G0
Ripple desired = AP
Freq. of passband edge = FP
Mag. error = G0/10^5
Ripple error = AP/10^5
Freq. error = FP/10^5
Max. iteration count = 50
iteration count = 0
Typflag = 2 (Chebyshev filter)
Choice = 1 (Warp to magnitude)
D
(page 45)
Find the following:
(FINDPK)
1) Gain at zero freq., EXPGO
2) Freq. of passband mag. edge, FLEAV
3) Freq. of reflection zero, FREFZ
4) Mag. at reflection zero, MAGREFZ.
ERROR FLAG = 1?
Yes
RETURN
No

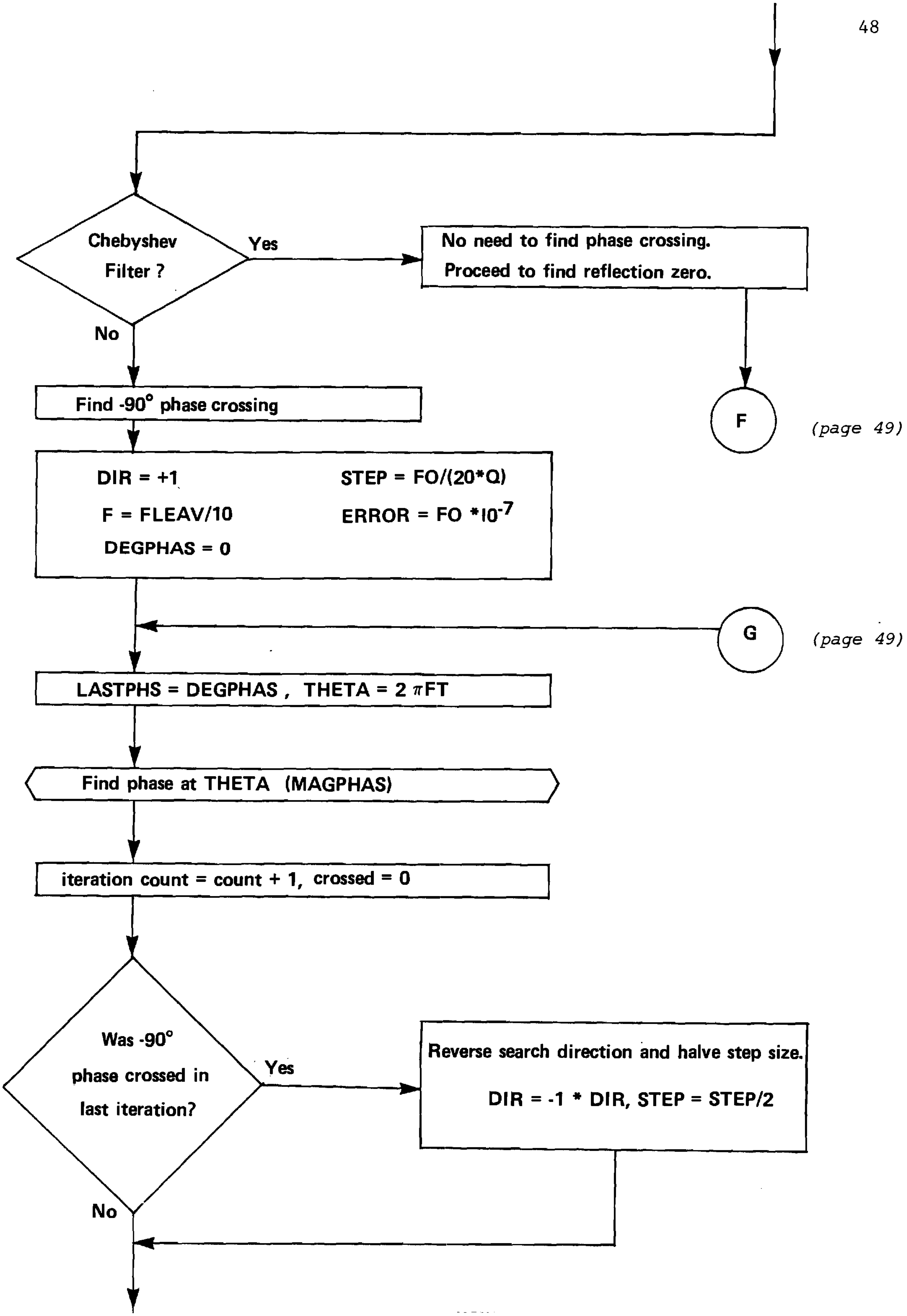

RUNFLAG = 0; iteration count = count + 1
Ripple deviation = (Ripple desired) — 20 log (MAGREFZ/EXPGO)
ABS. RIPPLE DEVIATION ≥ RIPPLE ERROR?
YES
AP = AP + (Ripple deviation)
Recalculate FO, BW, Q (CHEBSHV)
RUNFLAG = 1
NO
Gain deviation = (Gain desired) — EXPGO
ABS. GAIN DEVIATION ≥ MAG. ERROR?
YES
GO = GO + (Gain deviation)
RUNFLAG = 1
NO
Freq. deviation = (Freq. desired) — FLEAV

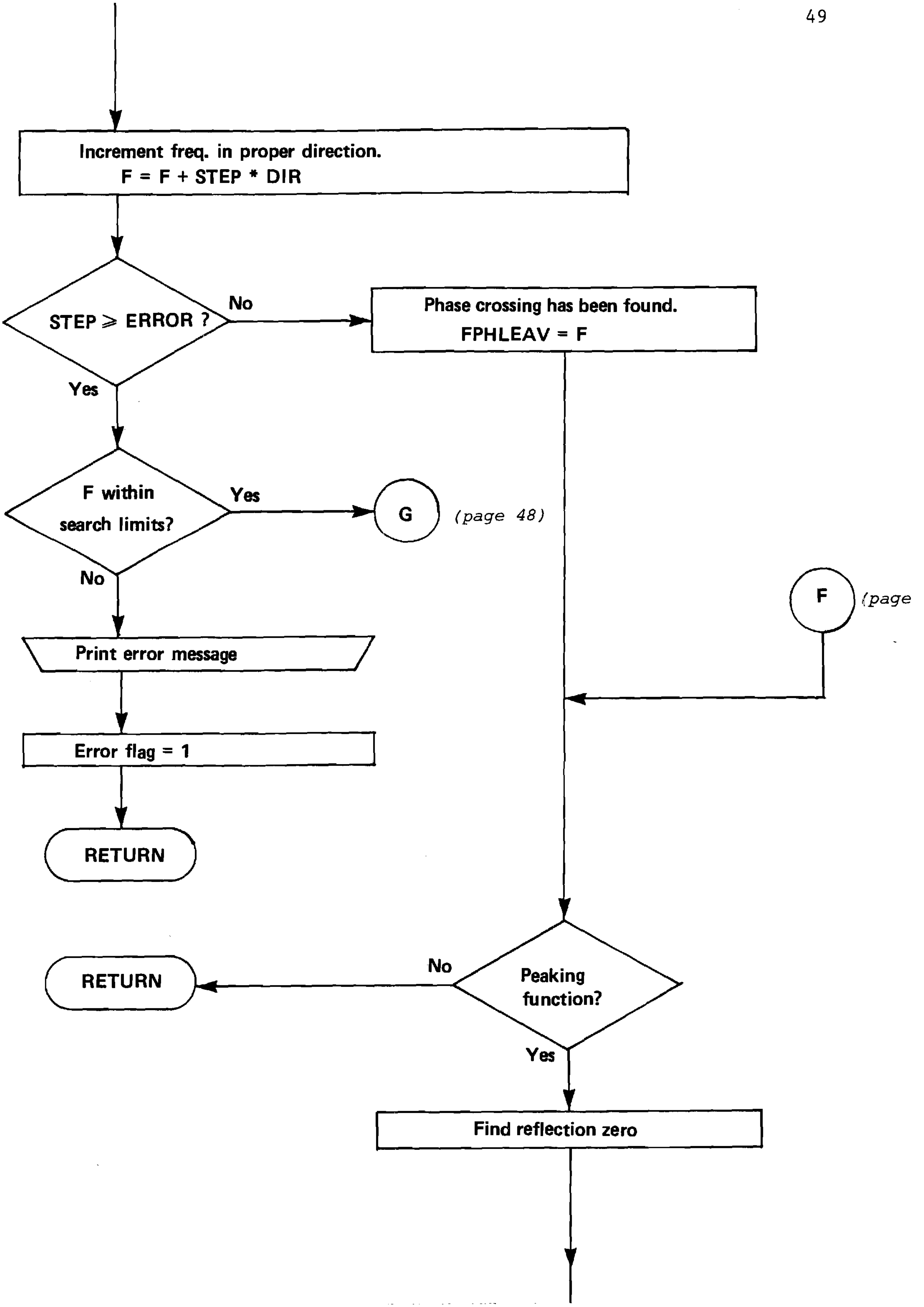

ABS. FREQ. DEVIATION < FREQ. ERROR?
No
Yes
FP = FP + (Freq. deviation)
Recalculate FO, BW, Q
(CHEBSHV)
RUNFLAG = 0?
No
Yes
RUNFLAG = 1
RUNFLAG = 0?
Yes
RETURN
No
ITERATION COUNT ≤ MAX. RUNS?
Yes
D
(page 43)
No
Print "Maxruns Exceeded"
Set error flag = 1
RETURN

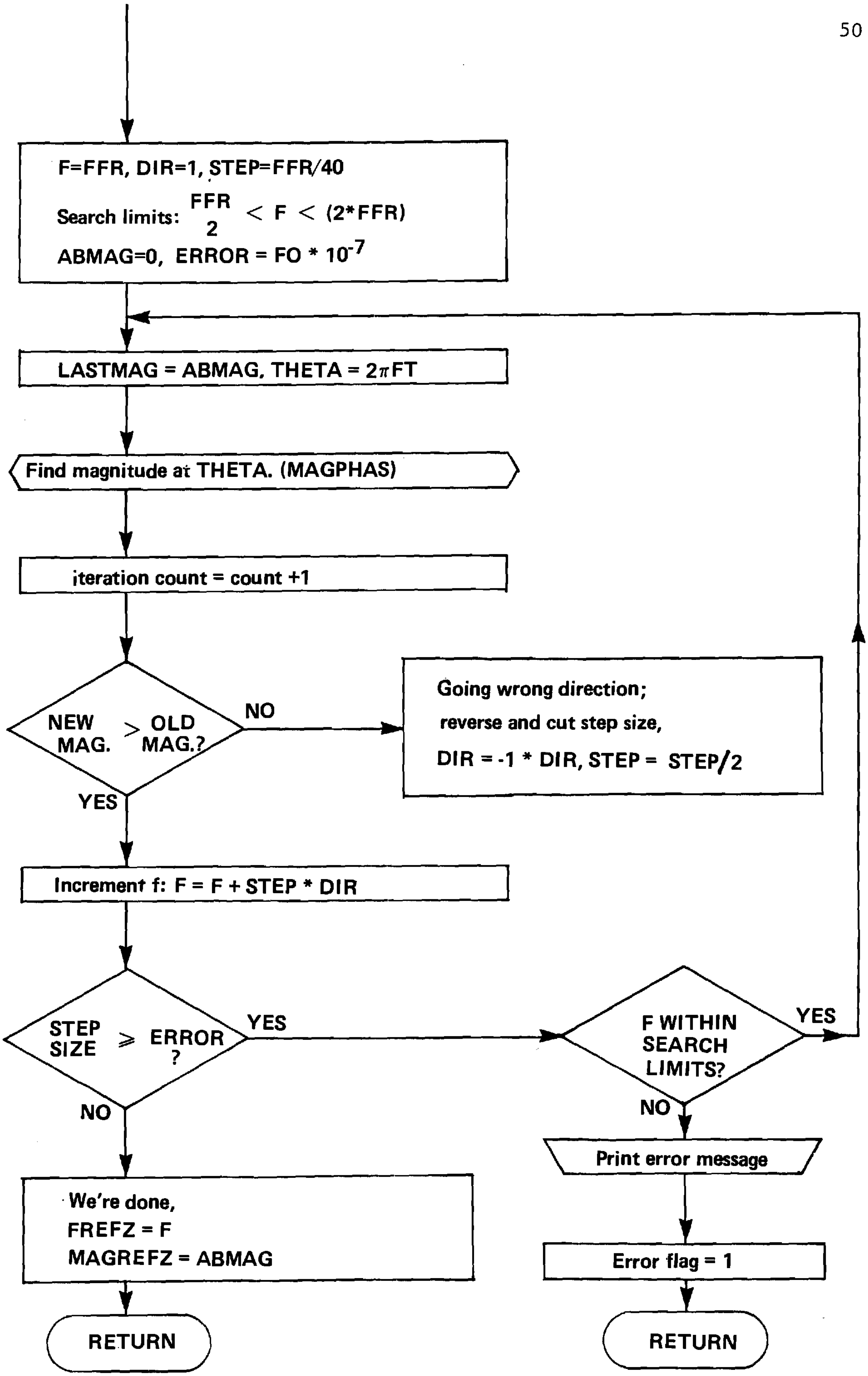
Subroutine Findpk



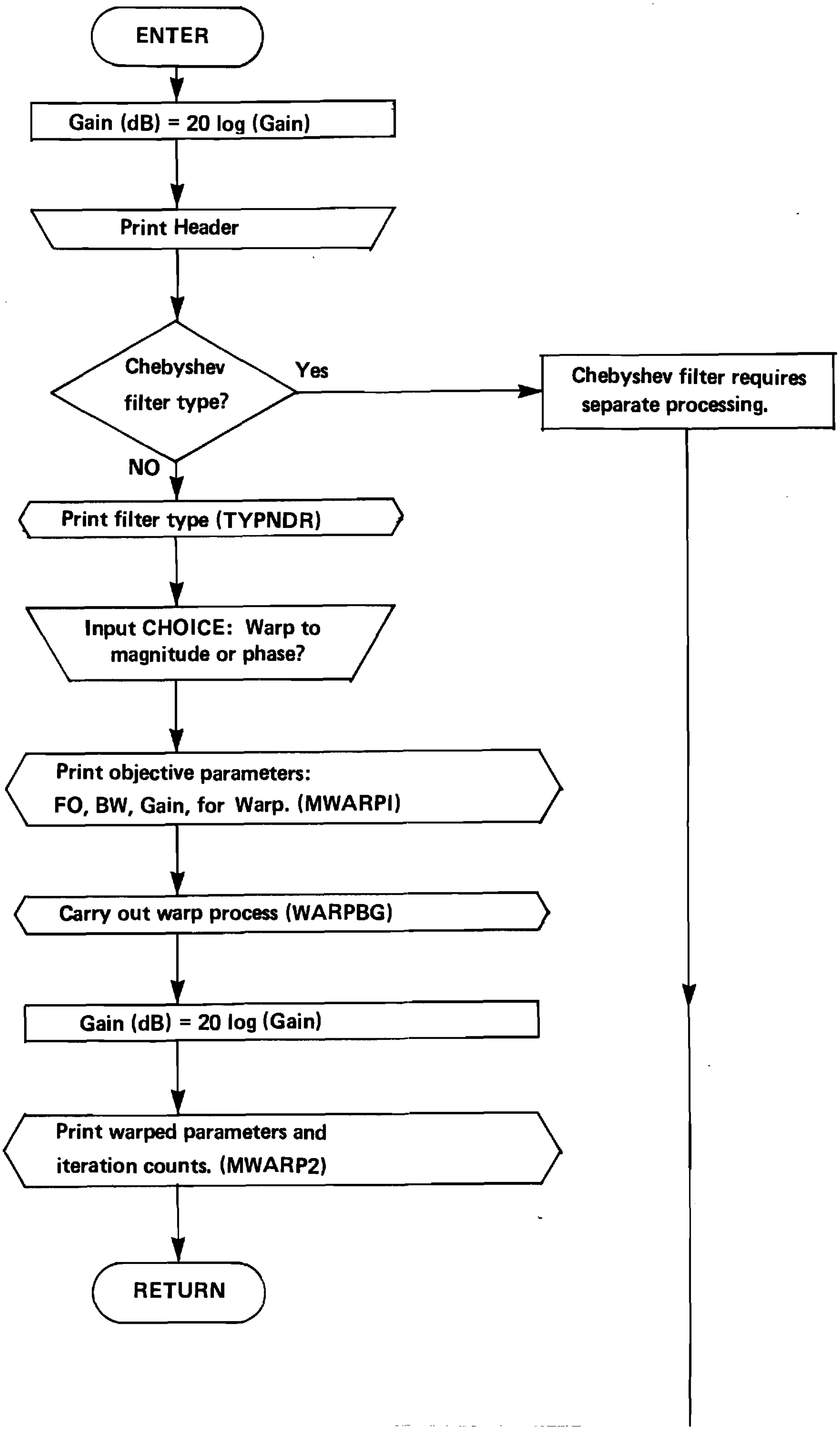

E
(below)
LASTMAG = ABMAG
THETA = 2 πFT
Find mag. + phase at THETA (MAGPHAS)
iteration count = count + 1 , crossed = 0
Was passband edge magnitude crossed in last iteration?
Yes
Reverse search direction and halve step size.
DIR = -1 * DIR, STEP = STEP/2
No
Increment freq. in proper direction.
F = F + STEP * DIR
STEP ≥ ERROR?
No
Freq. of passband edge has been found .
FLEAV = F
Yes
F within search limits?
No
Print error message
Error flag = 1
RETURN
Yes
E
(above)

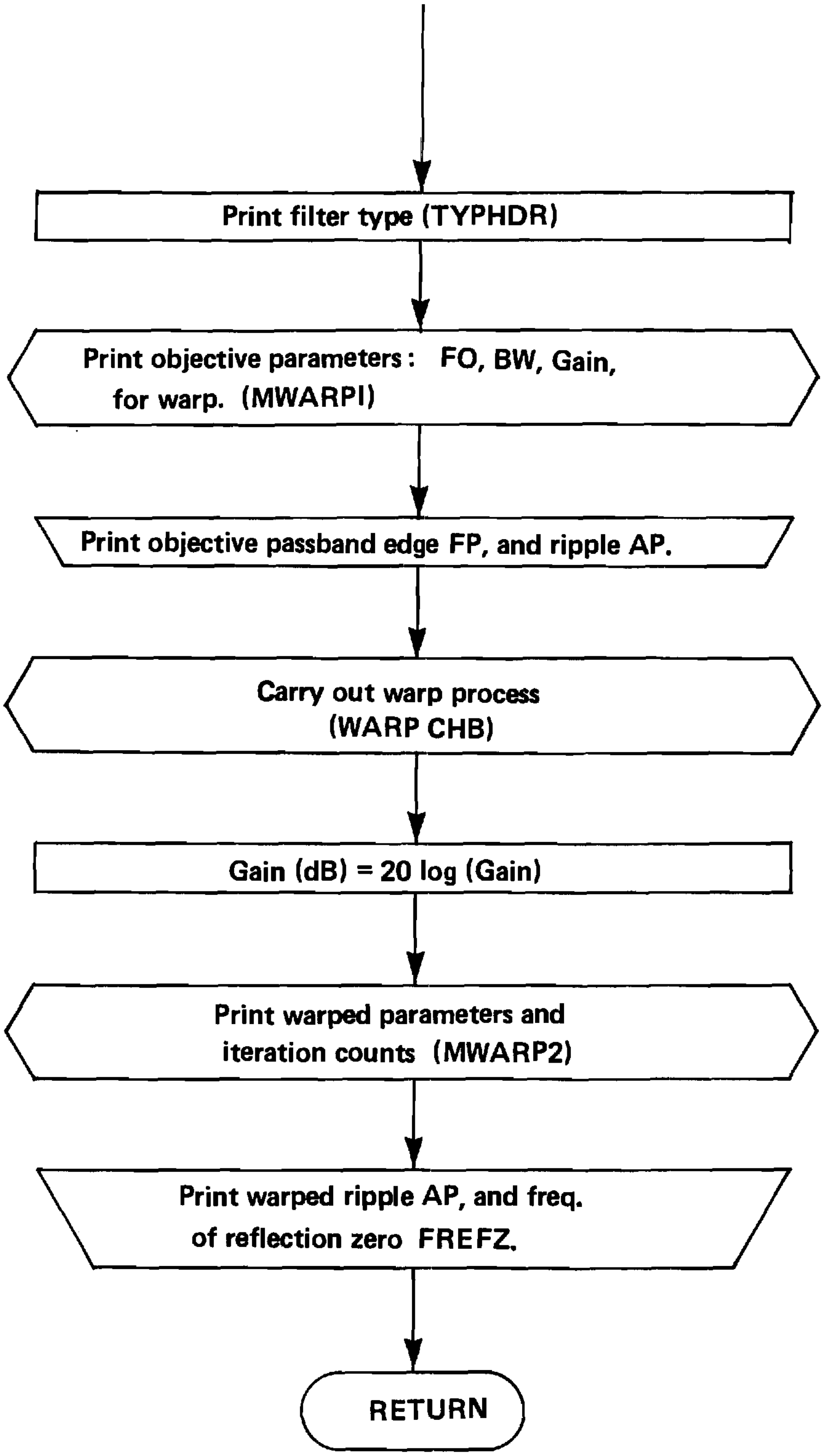

Chebyshev Filter ?
Yes
No need to find phase crossing. Proceed to find reflection zero.
F
(page 49)
No
Find -90° phase crossing
DIR = +1
STEP = FO/(20*Q)
F = FLEAV/10
ERROR = FO *10^-7
DEGPHAS = 0
G
(page 49)
LASTPHS = DEGPHAS , THETA = 2 πFT
Find phase at THETA (MAGPHAS)
iteration count = count + 1, crossed = 0
Was -90° phase crossed in last iteration?
Yes
Reverse search direction and halve step size.
DIR = -1 * DIR, STEP = STEP/2
No


Increment freq. in proper direction.
F = F + STEP * DIR
STEP ≥ ERROR ?
No
Phase crossing has been found.
FPHLEAV = F
Yes
F within search limits?
Yes
G
(page 48)
No
F
(page 4
Print error message
Error flag = 1
RETURN
RETURN
No
Peaking function?
Yes
Find reflection zero


F=FFR, DIR=1, STEP=FFR/40
Search limits: FFR/2 < F < (2*FFR)
ABMAG=0, ERROR = FO * 10-7
LASTMAG = ABMAG, THETA = 2πFT
Find magnitude at THETA. (MAGPHAS)
iteration count = count +1
NEW MAG. > OLD MAG.?
NO
Going wrong direction;
reverse and cut step size,
DIR = -1 * DIR, STEP = STEP/2
YES
Increment f: F = F + STEP * DIR
STEP SIZE ≥ ERROR ?
YES
F WITHIN SEARCH LIMITS?
YES
NO
NO
We're done,
FREFZ = F
MAGREFZ = ABMAG
RETURN
Print error message
Error flag = 1
RETURN

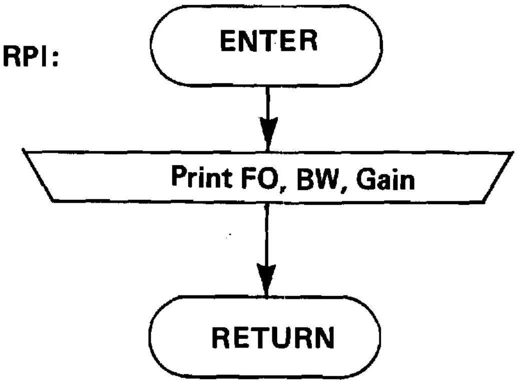
Subroutine Mwarp



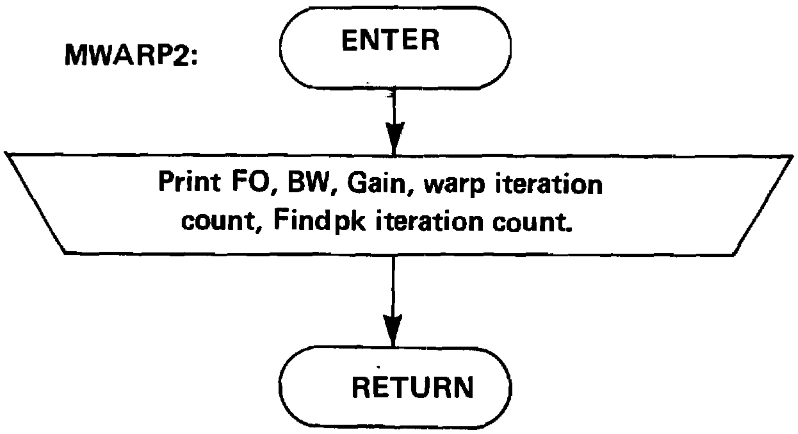

Print filter type (TYPHDR)
Print objective parameters: FO, BW, Gain, for warp. (MWARPI)
Print objective passband edge FP, and ripple AP.
Carry out warp process (WARP CHB)
Gain (dB) = 20 log (Gain)
Print warped parameters and iteration counts (MWARP2)
Print warped ripple AP, and freq. of reflection zero FREFZ.
RETURN

Subroutines Mwarpl and Mwarp 2

MWARPI:



MWARP2:



The following subroutines are common to all the computer-aided design programs. These subroutines are identified as:

SPECTRM

SWEEP

OPTIMIZ

MAGPHAS

FRESP

ALPHAS

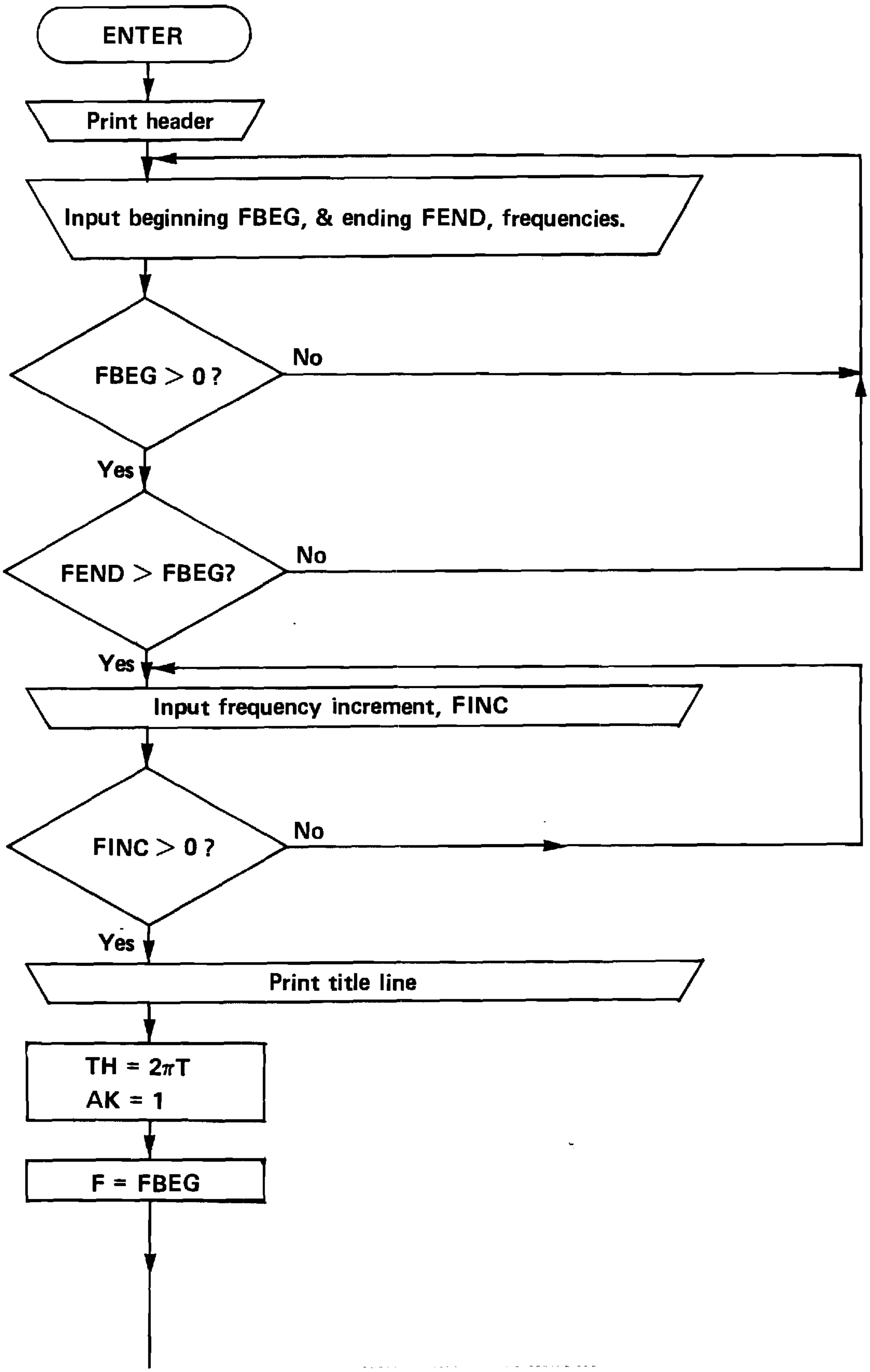
PRECAPS

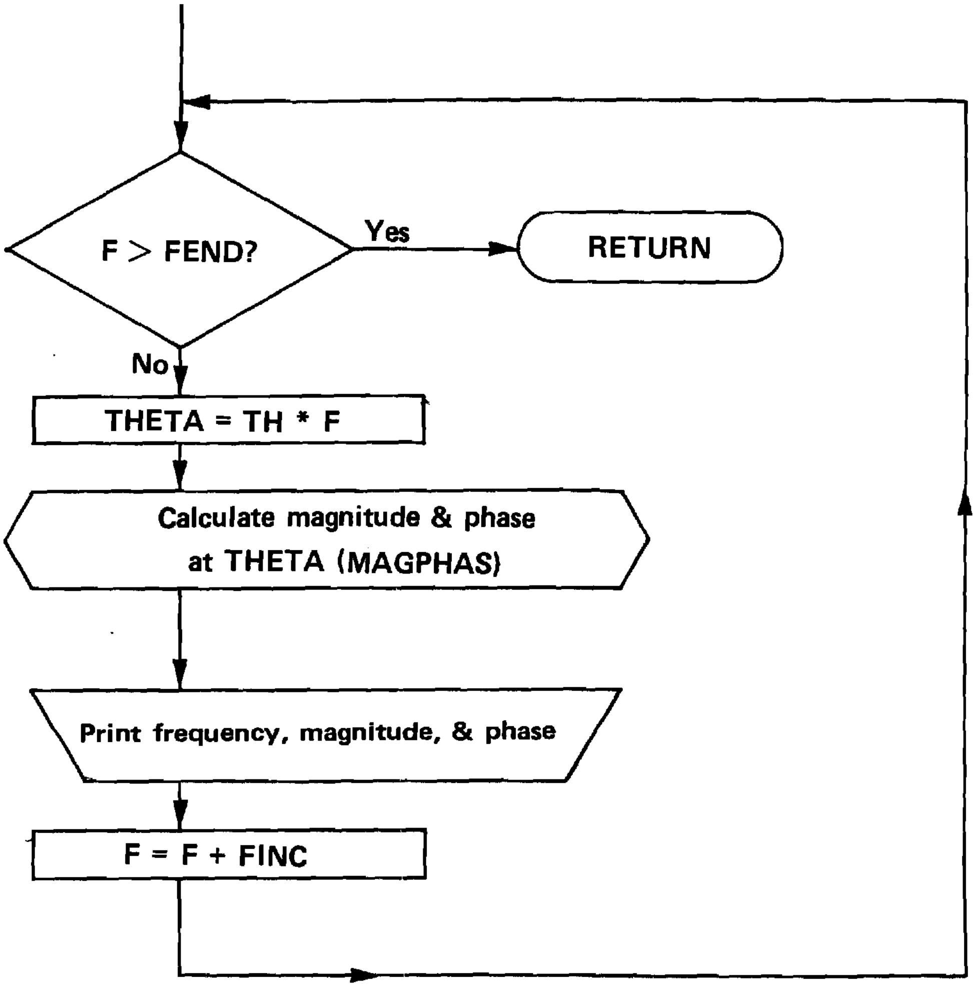
CAPS

POSTCAP

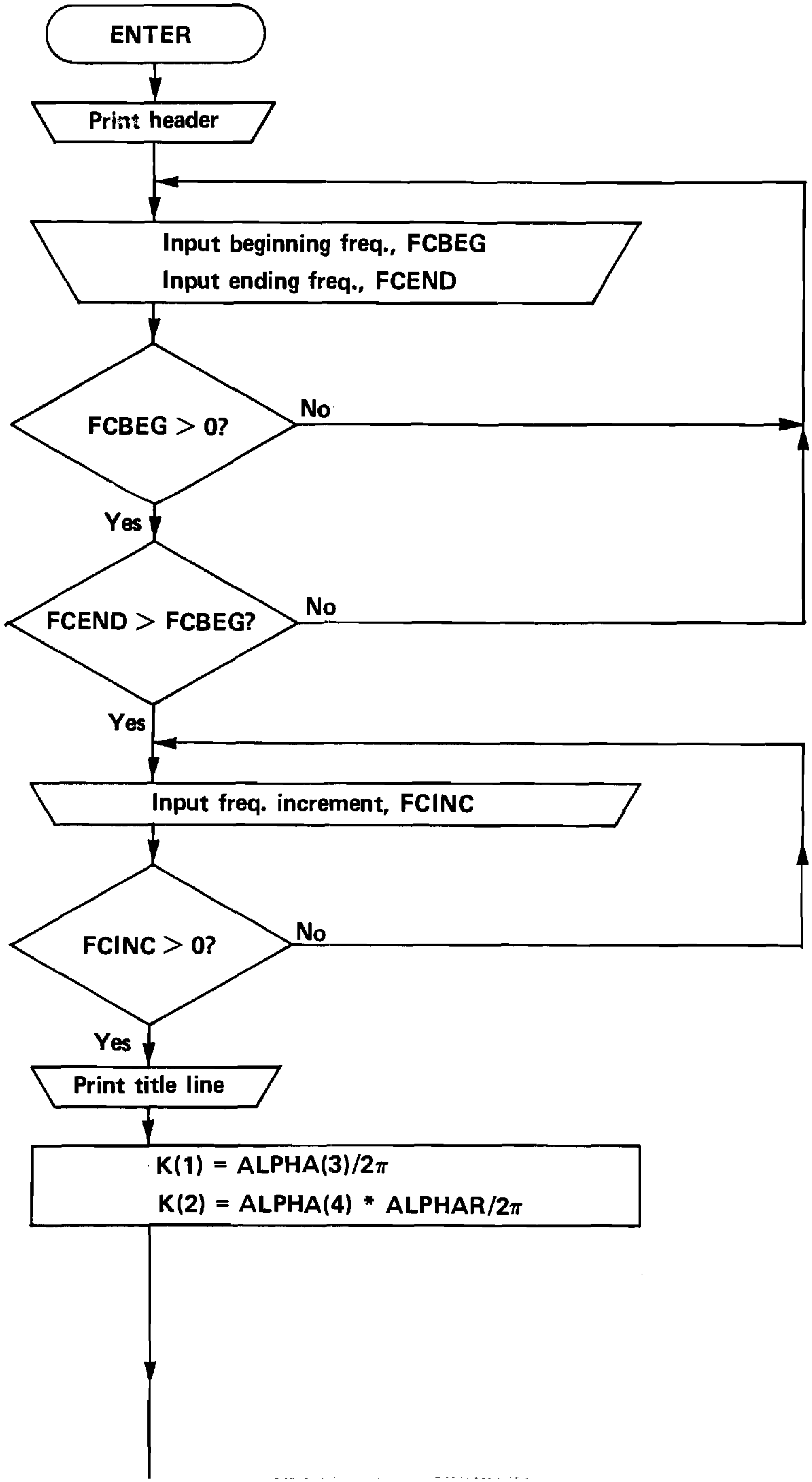
PRINT

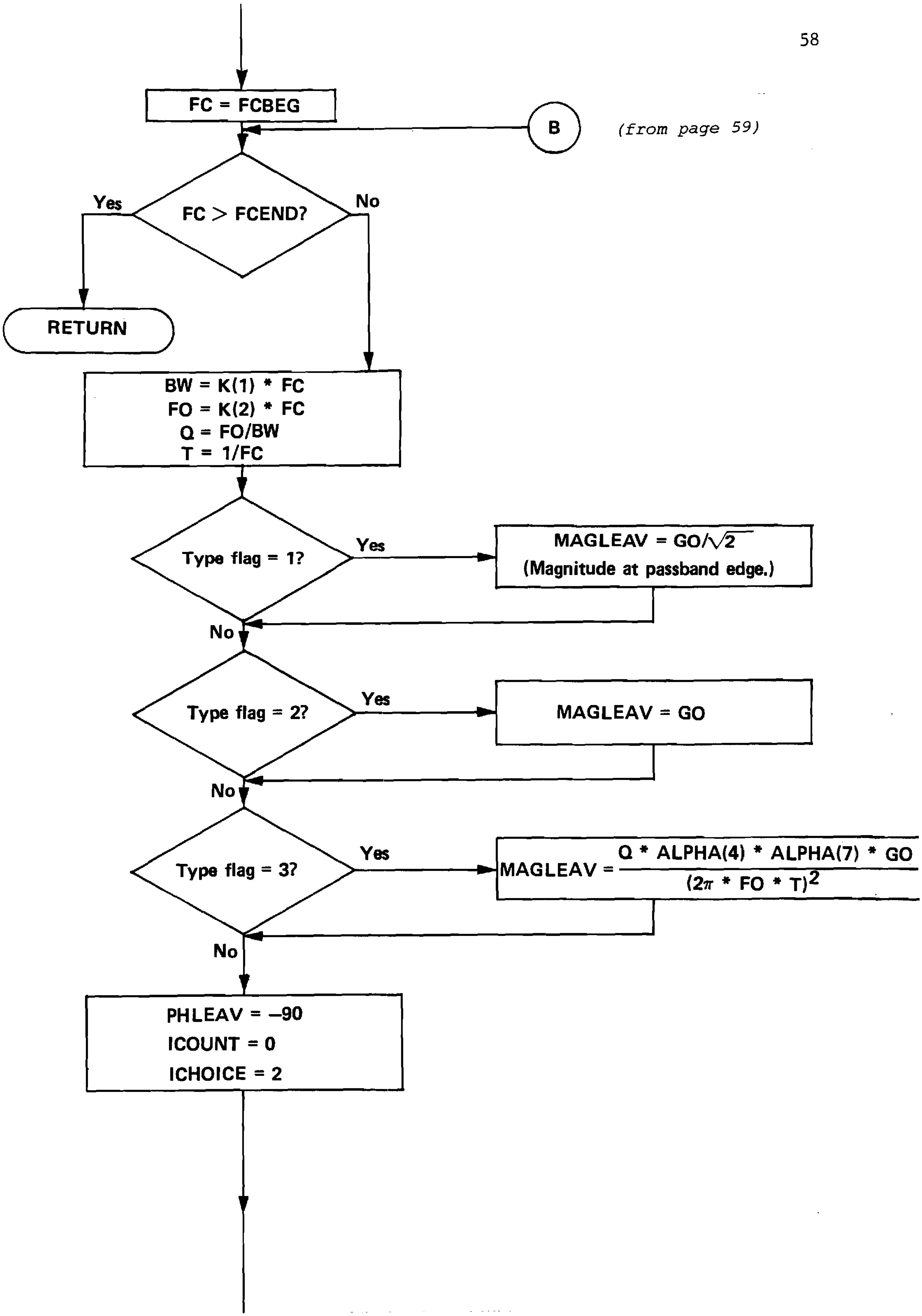
Subroutine Spectrum (MARTLP)

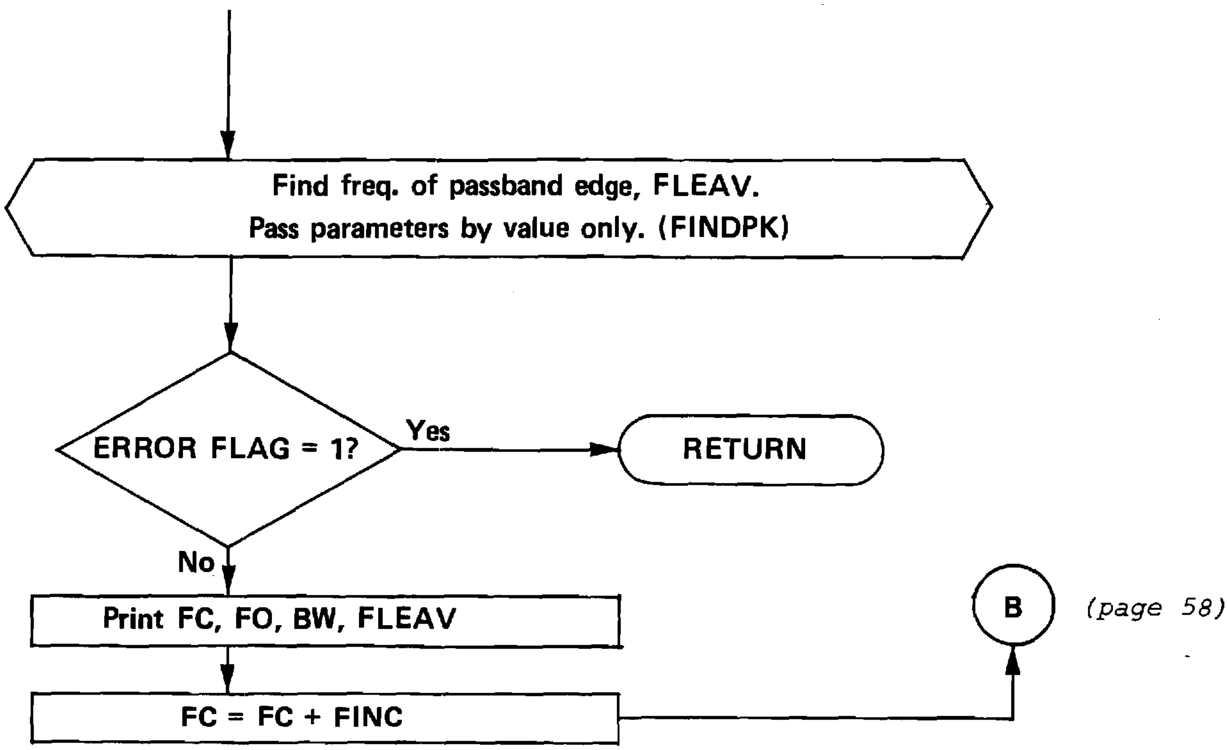



F > FEND?
Yes
RETURN
No
THETA = TH * F
Calculate magnitude & phase
at THETA (MAGPHAS)
Print frequency, magnitude, & phase
F = F + FINC

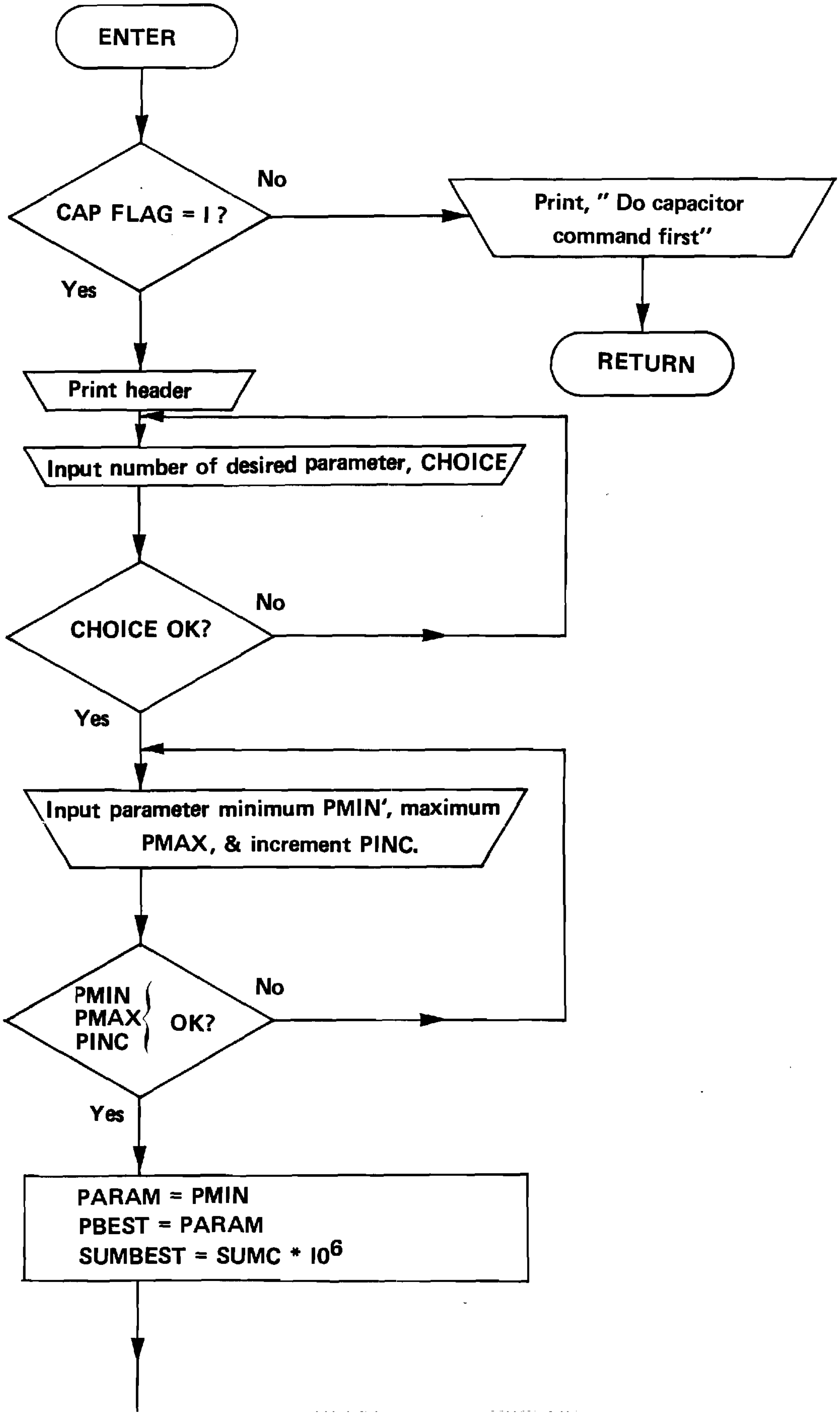
Subroutine Sweep (MARTLP)

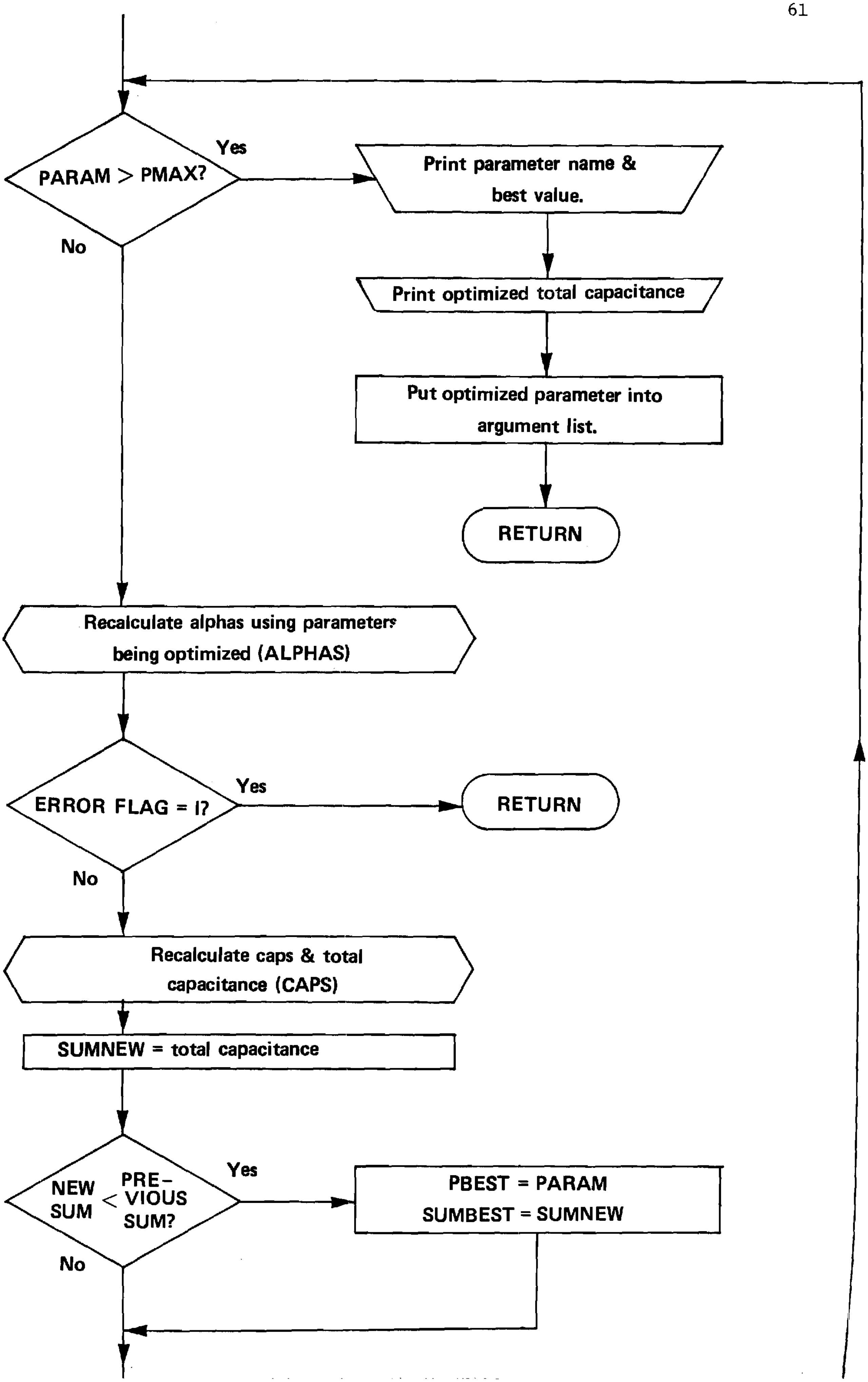


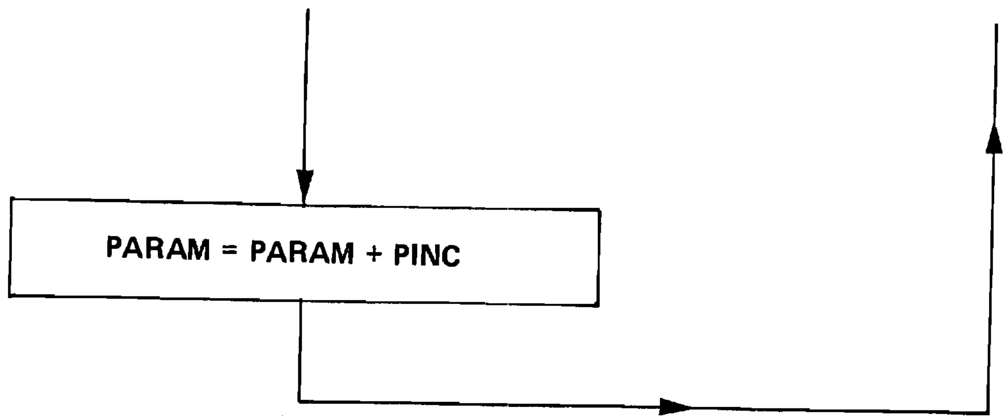

FC = FCBEG
B
(from page 59)
FC > FCEND?
Yes
No
RETURN
BW = K(1) * FC
FO = K(2) * FC
Q = FO/BW
T = 1/FC
Type flag = 1?
Yes
MAGLEAV = GO/√2
(Magnitude at passband edge.)
No
Type flag = 2?
Yes
MAGLEAV = GO
No
Type flag = 3?
Yes
MAGLEAV = Q * ALPHA(4) * ALPHA(7) * GO / (2π * FO * T)^2
No
PHLEAV = –90
ICOUNT = 0
ICHOICE = 2


Find freq. of passband edge, FLEAV.
Pass parameters by value only. (FINDPK)
ERROR FLAG = 1?
Yes
RETURN
No
Print FC, FO, BW, FLEAV
FC = FC + FINC
B
(page 58)

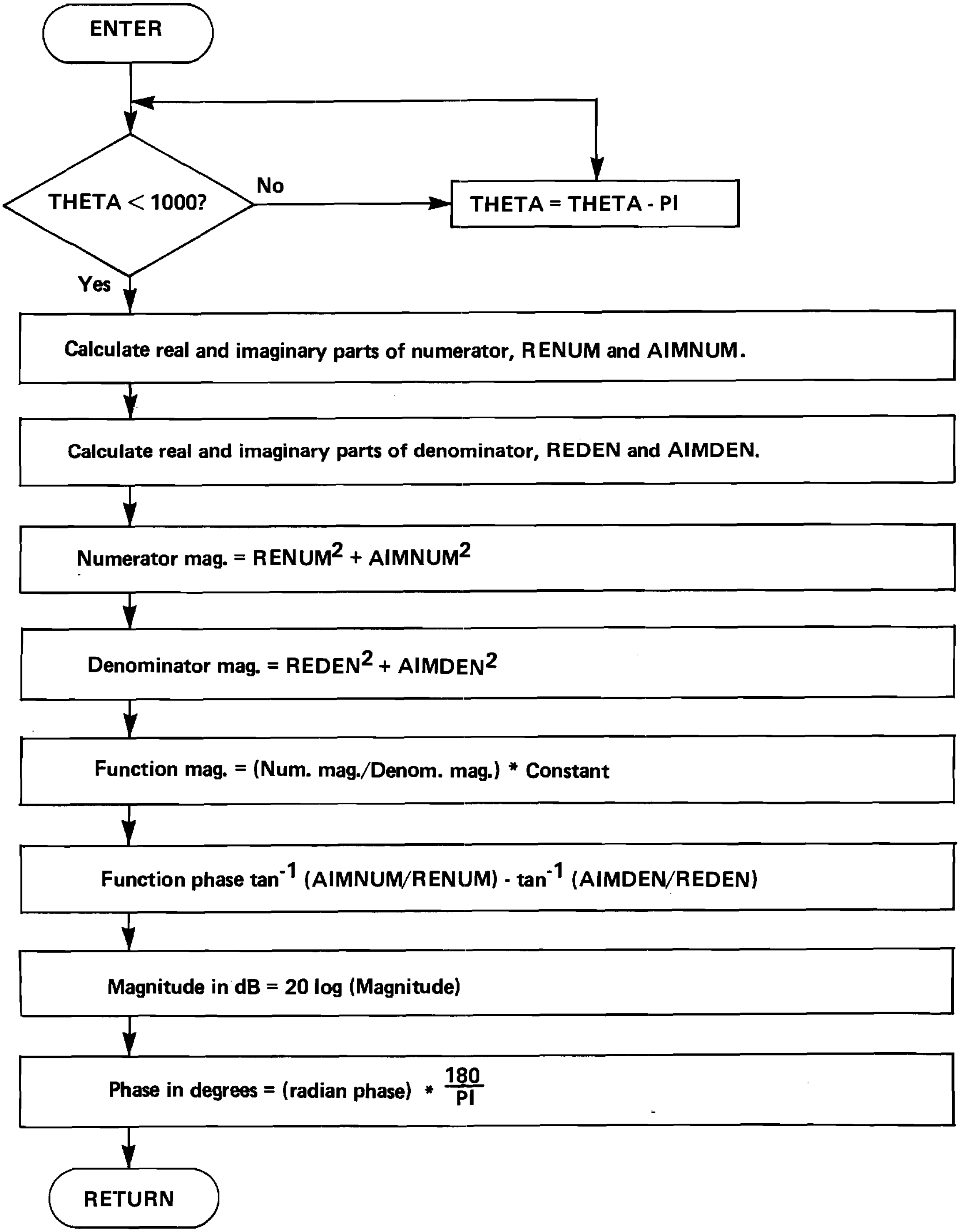
Subroutine Optimiz (MARTLP)



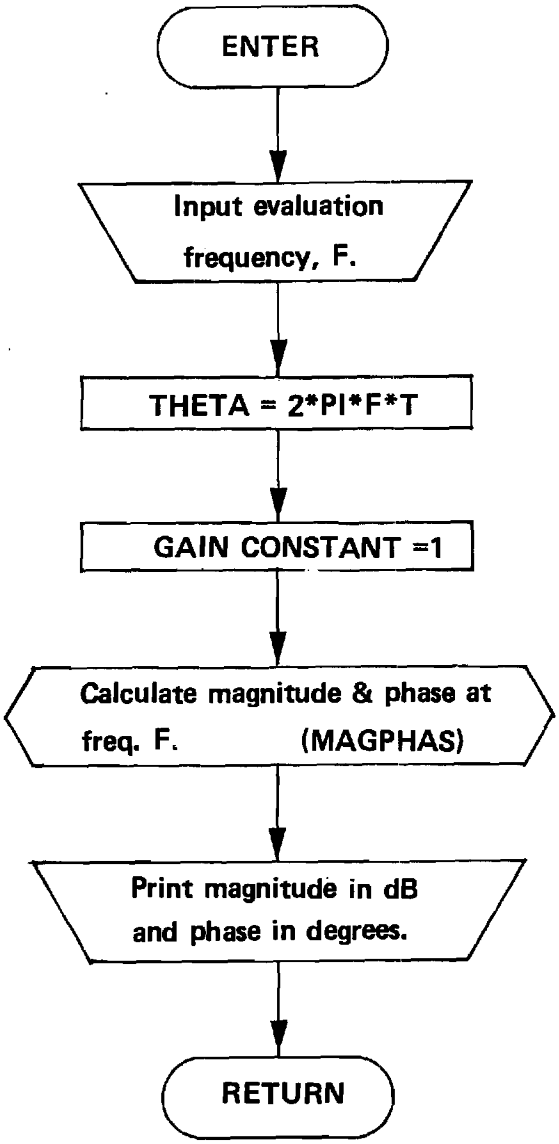

PARAM > PMAX?
Yes
No
Print parameter name & best value.
Print optimized total capacitance
Put optimized parameter into argument list.
RETURN
Recalculate alphas using parameters being optimized (ALPHAS)
ERROR FLAG = 1?
Yes
No
RETURN
Recalculate caps & total capacitance (CAPS)
SUMNEW = total capacitance
NEW SUM < PREVIOUS SUM?
Yes
No
PBEST = PARAM
SUMBEST = SUMNEW

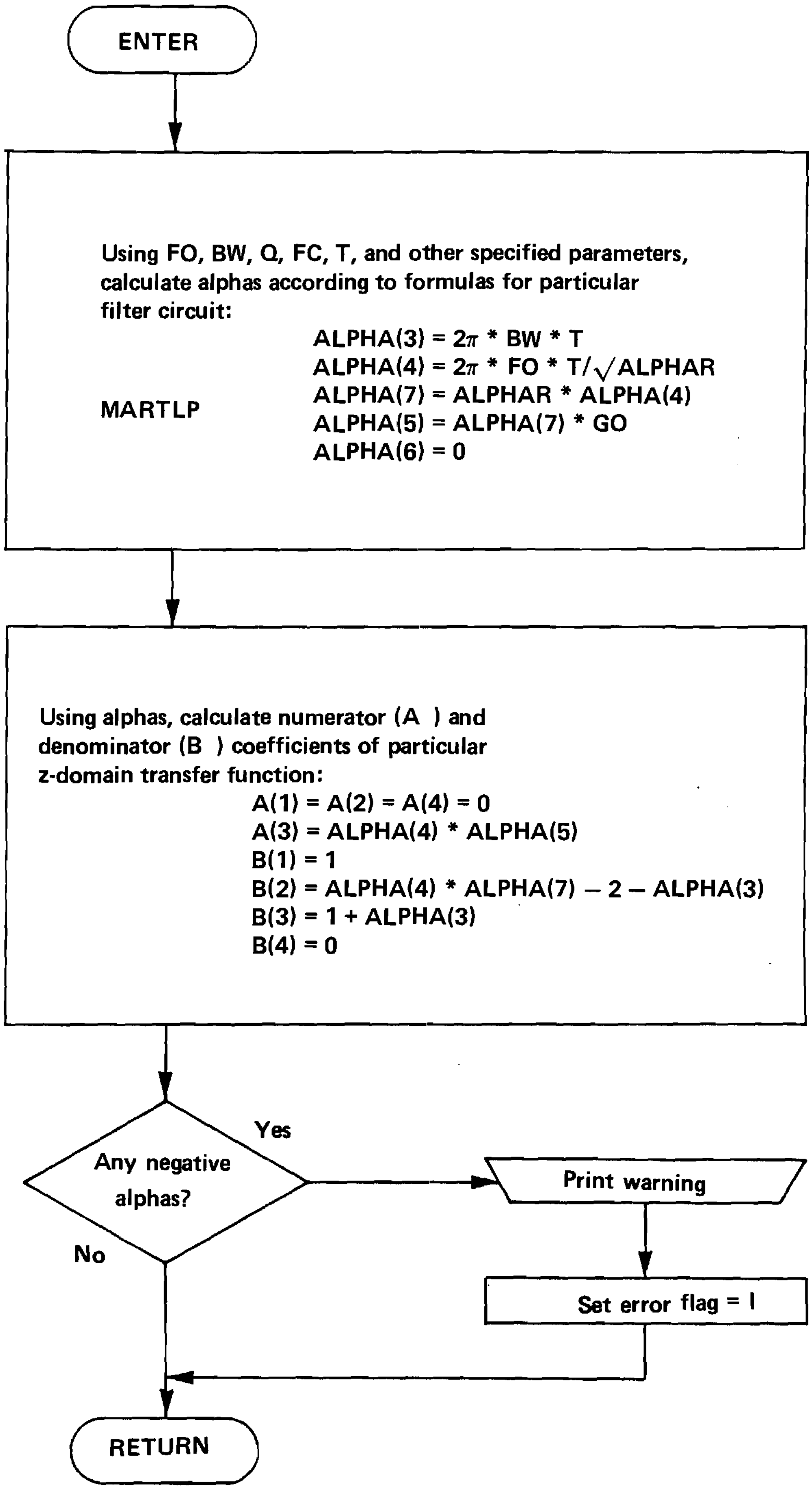

PARAM = PARAM + PINC

Subroutine Magphas

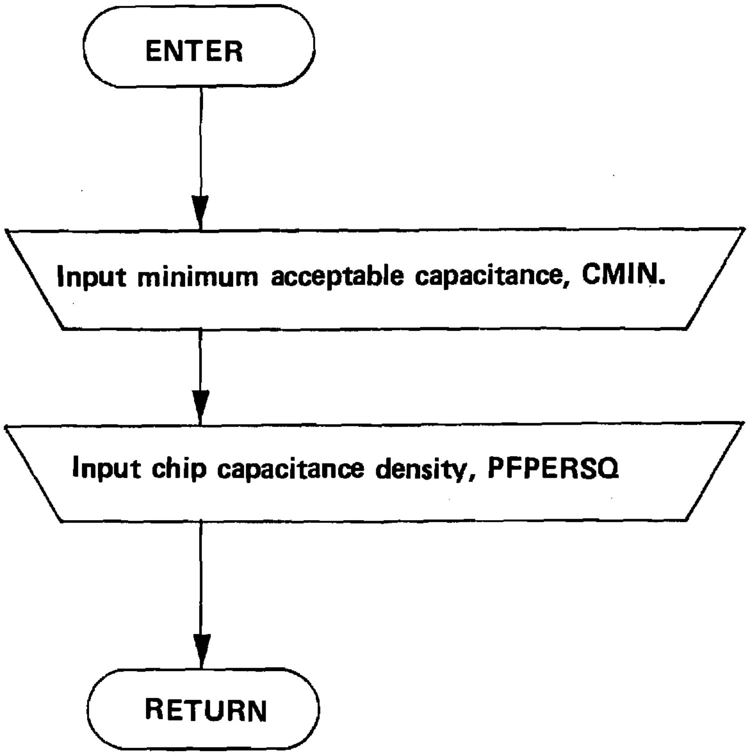


Subroutine Fresp

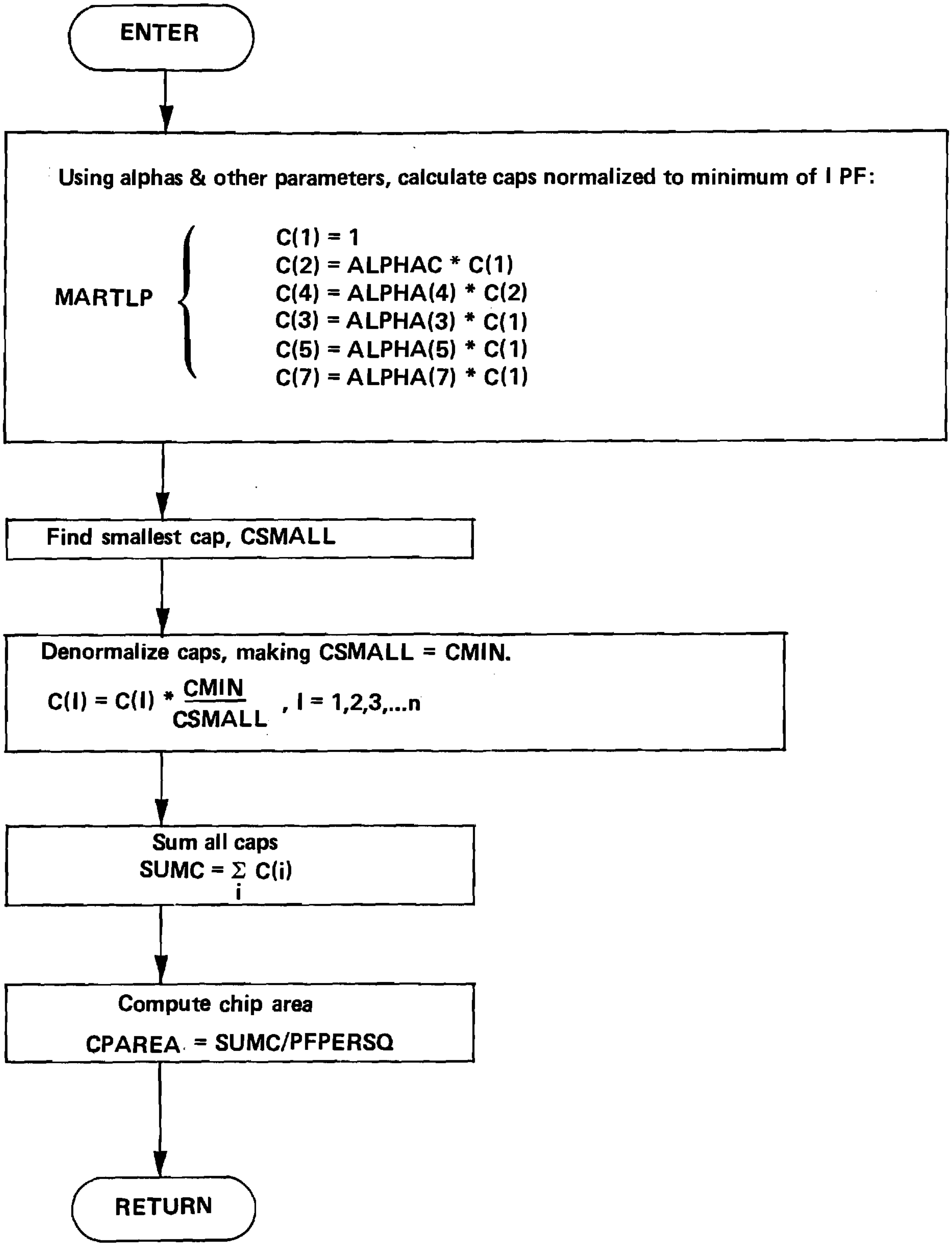



ENTER
Using FO, BW, Q, FC, T, and other specified parameters, calculate alphas according to formulas for particular filter circuit:
MARTLP
ALPHA(3) = 2π * BW * T
ALPHA(4) = 2π * FO * T/√ALPHAR
ALPHA(7) = ALPHAR * ALPHA(4)
ALPHA(5) = ALPHA(7) * GO
ALPHA(6) = 0
Using alphas, calculate numerator (A) and denominator (B) coefficients of particular z-domain transfer function:
A(1) = A(2) = A(4) = 0
A(3) = ALPHA(4) * ALPHA(5)
B(1) = 1
B(2) = ALPHA(4) * ALPHA(7) – 2 – ALPHA(3)
B(3) = 1 + ALPHA(3)
B(4) = 0
Any negative alphas?
Yes
No
Print warning
Set error flag = I
RETURN

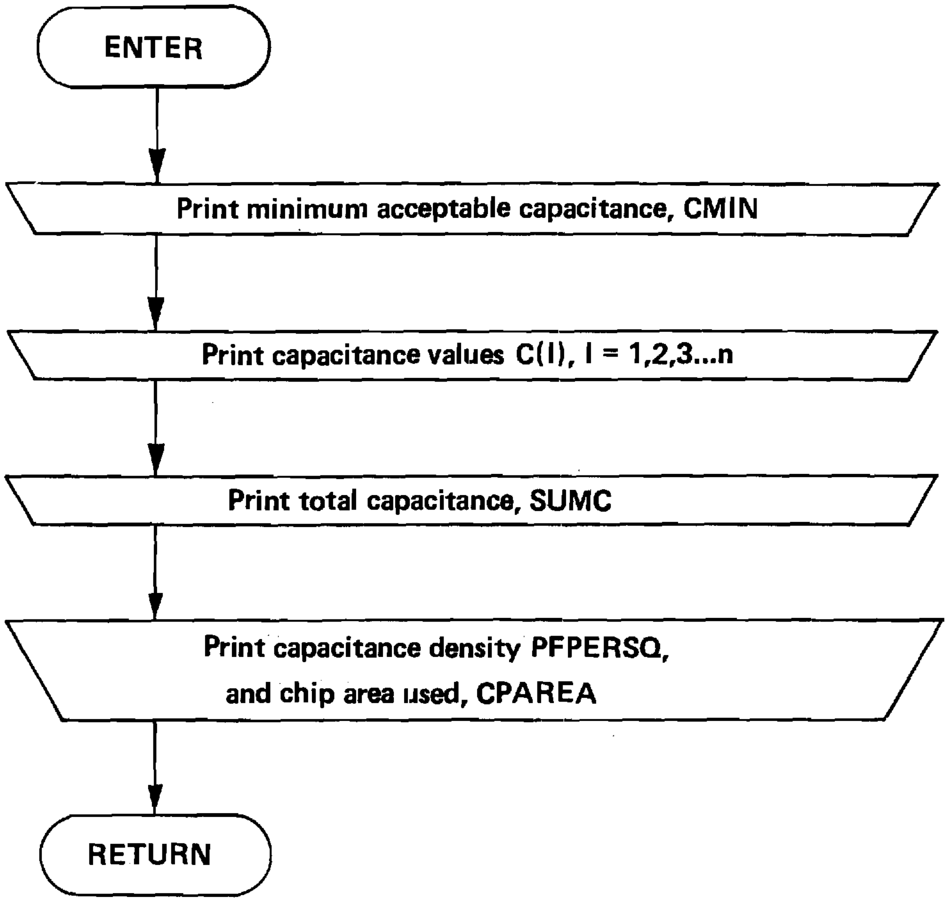
Subroutine Precaps



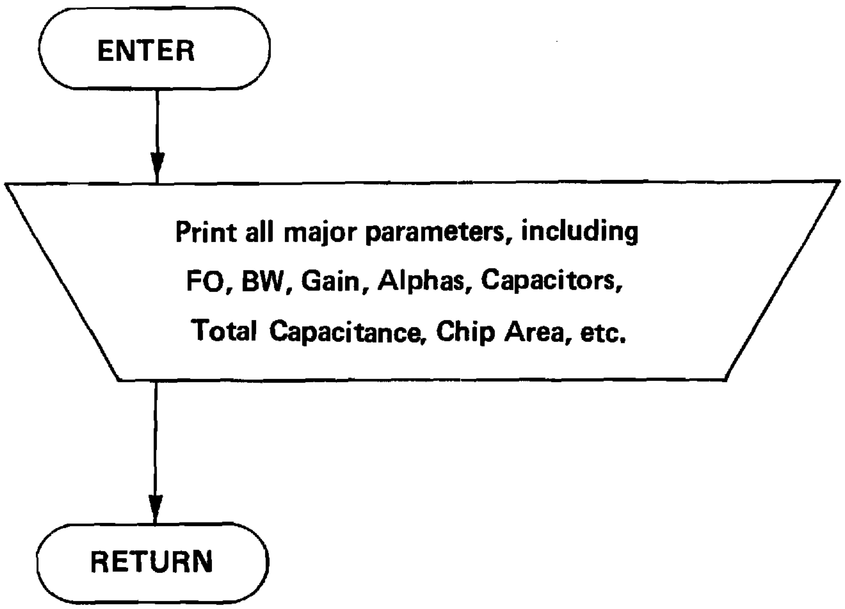
Subroutine Caps



Subroutine Postcap



Subroutine Print



E. Sample Low-pass Design Using MARTLP

```
*OLD MARTLP
*FORTRAN
*RUN-10 MARTLP
SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR
MARTIN LOWPASS FILTER.
FORTRAN VERSION BY W.N. GOOLSBY.
REVISION 6/5/81.

NEED MENU?  (0=NO, 1=YES)
=1
A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,
ENTER THE NUMBER OF THE APPROPRIATE COMMAND.

1.  ENTER PARAMETERS FOR BUTTERWORTH LOWPASS FILTER
2.  ENTER PARAMETERS FOR CHEBYSHEV LOWPASS FILTER
3.  ENTER PARAMETERS FOR GENERALIZED LOWPASS FILTER
4.  PRINT MENU
5.  TERMINATE PROGRAM
6.  WARP FO, BW, AND GAIN.
7.  PRINT CURRENT VALUES.
8.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F
9.  FIND CAPACITOR VALUES AND TOTAL C
10.  CHANGE FO ONLY
11.  CHANGE CHEBYSHEV RIPPLE WIDTH ONLY
12.  CHANGE GENERALIZED FILTER BANDWIDTH ONLY
13.  CHANGE FC ONLY
14.  CHANGE ALPHAC=C(2)/C(1) ONLY
15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4) ONLY
16.  CHANGE GAIN AT F=0 ONLY (DECIBELS)
17.  CHANGE CHEBYSHEV PASSBAND EDGE, FP, ONLY.
18.  FIND MAGNITUDE AND PHASE AT A SERIES
     OF FREQUENCY POINTS.
19.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON FO & BW
20.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER

ENTER COMMAND NUMBER (#4 FOR MENU)
=1
FILTER TYPE:  BUTTERWORTH
ENTER POLE FREQUENCY, FO (HERTZ)
=3000
ENTER GAIN AT F=0, DBGO (DECIBELS)
=6
ENTER CLOCK FREQUENCY, FC (HERTZ)
=100000
ENTER ALPHAC=C2/C1
=1
ENTER ALPHAR=ALPHA(7)/ALPHA(4)
=1
```

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=9
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF
AND C2/C1=      1.00000000 ,THE CAPACITANCE VALUES ARE:

C(1)=   5.3051648      PF
C(2)=   5.3051648      PF
C(3)=   1.4142136      PF
C(4)=  1.00000000      PF
C(5)=   1.9952623      PF
C(7)=  1.00000000      PF

TOTAL CAPACITANCE USED=   16.019805      PF
AT  0.2000     PF/SQ, AREA USED =   80.0990      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=20
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  ALPHAR=ALPHA(7)/ALPHA(4)
=1

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAC
FOR PARAMETER=      1.00000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   16.019805      PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=20
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  ALPHAR=ALPHA(7)/ALPHA(4)
=2

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAR
FOR PARAMETER=      1.00000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   16.019805      PF
```

```
THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=18
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1


 FREQUENCY (HZ)    MAGNITUDE (DB)    MAGNITUDE        PHASE (DEG)

 2995.0000         3.0132958         1.4147014        -84.355480
 2996.0000         3.0108047         1.4142957        -84.380510
 2997.0000         3.0083127         1.4138900        -84.405533
 2998.0000         3.0058200         1.4134843        -84.430550
 2999.0000         3.0033265         1.4130786        -84.455560
 3000.0000         3.0008322         1.4126729        -84.480563
 3001.0000         2.9983371         1.4122671        -84.505559
 3002.0000         2.9958412         1.4118614        -84.530549
 3003.0000         2.9933445         1.4114556        -84.555531
 3004.0000         2.9908471         1.4110498        -84.580508
 3005.0000         2.9883488         1.4106440        -84.605477

ENTER COMMAND NUMBER (#4 FOR MENU)
=6
ADJUSTS FREQUENCY AND MAGNITUDE TO COMPENSATE
FOR Z-TRANSFORM WARPING.

FILTER TYPE:  BUTTERWORTH
YOU MAY CHOOSE TO WARP FO TO APPLICABLE
PHASE OR MAGNITUDE.
ENTER NUMBER OF DESIRED COMMAND.
     1.  MAGNITUDE
     2.  PHASE
=1
DESIRED POLE FREQUENCY, FO=   3000.0000      HERTZ
DESIRED POLE BANDWIDTH, BW=   4242.6407      HERTZ
DESIRED GAIN AT ZERO FREQ., DBGO=   6.0000000      DECIBELS

WARPED VARIABLES FOLLOW
FO=   2995.5411      HERTZ
BW=   4236.3348      HERTZ
DBGO=   1.9952623      DECIBELS

WARP ITERATION COUNT=     2  CYCLES
FINDPK ITERATION COUNT=   188  CYCLES

ENTER COMMAND NUMBER (#4 FOR MENU)
=18
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	3.0021538	1.4128878	-84.475392
2996.0000	2.9996548	1.4124814	-84.500432
2997.0000	2.9971550	1.4120750	-84.525466
2998.0000	2.9946544	1.4116685	-84.550493
2999.0000	2.9921531	1.4112620	-84.575513
3000.0000	2.9896509	1.4108555	-84.600527
3001.0000	2.9871479	1.4104490	-84.625533
3002.0000	2.9846442	1.4100425	-84.650533
3003.0000	2.9821397	1.4096360	-84.675526
3004.0000	2.9796343	1.4092295	-84.700512
3005.0000	2.9771282	1.4088229	-84.725492

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=2
FILTER TYPE: CHEBYSHEV
ENTER FP= FREQUENCY WHERE MAGNITUDE
CHARACTERISTIC LEAVES PASSBAND (HERTZ).
=3000
ENTER RIPPLE WIDTH, AP  (DECIBELS)
=2
ENTER GAIN AT F=0, DBG0 (DECIBELS)
=6
ENTER CLOCK FREQUENCY, FC (HERTZ)
=100000
ENTER ALPHAC=C2/C1
=1
ENTER ALPHAR=ALPHA(7)/ALPHA(4)
=1

ENTER COMMAND NUMBER (#4 FOR MENU)
=9
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF
AND C2/C1=       1.00000000 ,THE CAPACITANCE VALUES ARE:

C(1)=   6.5999706      PF
C(2)=   6.5999706      PF
C(3)=   1.0000000      PF
C(4)=   1.1286492      PF
C(5)=   2.2519512      PF
C(7)=   1.1286492      PF

TOTAL CAPACITANCE USED=   18.709191      PF
AT  0.2000     PF/SQ, AREA USED =   93.5460      SQ MILS
```

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=20
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
      1.  ALPHAC=C2/C1
      2.  ALPHAR=ALPHA(7)/ALPHA(4)
=1

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAC
FOR PARAMETER=       0.89000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =    17.859043       PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=20
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
      1.  ALPHAC=C2/C1
      2.  ALPHAR=ALPHA(7)/ALPHA(4)
=2

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAR
FOR PARAMETER=       0.79000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =    17.608836       PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=9
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=    1.0000000       PF
AND C2/C1=       0.89000000 ,THE CAPACITANCE VALUES ARE:

C(1)=    6.5999706       PF
C(2)=    5.8739738       PF
C(3)=    1.0000000       PF
C(4)=    1.1301483       PF
C(5)=    2.0015781       PF
C(7)=    1.0031654       PF
```

```
TOTAL CAPACITANCE USED=   17.608836      PF
AT  0.2000     PF/SQ, AREA USED =   88.0442      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=18
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	6.2096573	2.0440093	-96.818585
2996.0000	6.2060454	2.0431595	-96.859648
2997.0000	6.2024303	2.0423093	-96.900690
2998.0000	6.1988121	2.0414587	-96.941713
2999.0000	6.1951906	2.0406077	-96.982715
3000.0000	6.1915660	2.0397564	-97.023698
3001.0000	6.1879382	2.0389046	-97.064661
3002.0000	6.1843073	2.0380525	-97.105604
3003.0000	6.1806732	2.0372000	-97.146526
3004.0000	6.1770360	2.0363471	-97.187429
3005.0000	6.1733957	2.0354938	-97.228312

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=6
ADJUSTS FREQUENCY AND MAGNITUDE TO COMPENSATE
FOR Z-TRANSFORM WARPING.

FILTER TYPE: CHEBYSHEV
DESIRED POLE FREQUENCY, FO=   2721.6803      HERTZ
DESIRED POLE BANDWIDTH, BW=   2411.4493      HERTZ
DESIRED GAIN AT ZERO FREQ., DBGO=   6.0000000      DECIBELS
DESIRED PASSBAND EDGE FP=   3000.0000      HERTZ
DESIRED RIPPLE WIDTH AP=   2.0000000      DB

WARPED VARIABLES FOLLOW
FO=   2616.5153      HERTZ
BW=   2240.1647      HERTZ
DBGO=   6.0000000      DECIBELS

WARP ITERATION COUNT=    5  CYCLES
FINDPK ITERATION COUNT=   483  CYCLES
CHEBYSHEV RIPPLE WIDTH, AP=   2.2279963      DB
REFLECTION ZERO OCCURS AT   2119.7474      HERTZ

ENTER COMMAND NUMBER (#4 FOR MENU)
=9
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
```

```
FOR MINIMUM C=    1.0000000       PF
AND C2/C1=       0.89000000 ,THE CAPACITANCE VALUES ARE:

C(1)=   7.1046091      PF
C(2)=   6.3231021      PF
C(3)=   1.0000000      PF
C(4)=   1.1695527      PF
C(5)=   2.0713662      PF
C(7)=   1.0381423      PF

TOTAL CAPACITANCE USED=   18.706772      PF
AT  0.2000      PF/SQ, AREA USED =   93.5339      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=18
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	6.0212694	2.0001542	-102.41285
2996.0000	6.0170212	1.9991761	-102.45423
2997.0000	6.0127701	1.9981979	-102.49558
2998.0000	6.0085160	1.9972195	-102.53690
2999.0000	6.0042590	1.9962409	-102.57820
3000.0000	5.9999992	1.9952621	-102.61947
3001.0000	5.9957365	1.9942832	-102.66072
3002.0000	5.9914708	1.9933040	-102.70194
3003.0000	5.9872024	1.9923247	-102.74314
3004.0000	5.9829310	1.9913452	-102.78431
3005.0000	5.9786568	1.9903655	-102.82545

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=5
```

IV. BANDPASS FILTER

A. Design Approach

A resonator active filter with bandpass and low-pass responses was shown in Fig. 3.1, and its transfer function was shown to be

$$\frac{V_{BP}}{V_{in}} = -\frac{s/R_5C_1}{s^2 + s/R_3C_1 + 1/R_4C_2R_7C_1} \tag{4-1}$$

The peak gain is

$$G = R_3/R_5 \tag{4-2}$$

The center frequency is

$$f_o = \frac{\omega_o}{2\pi} = \frac{1}{2\pi\sqrt{R_4C_2R_7C_1}} \tag{4-3}$$

The bandwidth is

$$BW = \frac{B}{2\pi} = \frac{1}{2\pi\ R_3C_1} \tag{4-4}$$

The filter Q is

$$Q = \frac{f_o}{BW} = R_3\sqrt{\frac{C_1}{R_4R_7C_2}} \tag{4-5}$$

The design equations are:

$$\alpha_4 = 2\pi f_o\ T_C/\sqrt{\alpha_R} \tag{4-6}$$

$$\alpha_7 = \alpha_R \alpha_4 \tag{4-7}$$

$$\alpha_3 = 2\pi \; BW \; T_C \tag{4-8}$$

$$\alpha_5 = \alpha_3 G \tag{4-9}$$

The sensitivities are:

$$S_{\alpha_4}^{f_o} = \frac{\partial f_o / f_o}{\partial \alpha_4 / \alpha_4} = 1 \tag{4-10}$$

$$S_{\alpha_7}^{f_o} = 1 \tag{4-11}$$

$$S_{\alpha_3}^{BW} = 1 \tag{4-12}$$

$$S_{\alpha_5}^{G} = 1 \tag{4-13}$$

$$S_{\alpha_3}^{G} = -1 \tag{4-14}$$

The Z-transformation for the bandpass filter was obtained from the block diagram shown in Fig. 3.3(b) as

$$\frac{V_{BP}}{V_{in}} = \frac{-\alpha_5 (Z - 1)}{Z^2 + Z(\alpha_3 - 2 + \alpha_4 \alpha_7) + 1 - \alpha_3} \tag{4-15}$$

$$= \frac{A_1 Z + A_o}{B_2 Z^2 + B_1 Z + B_o} \tag{4-16}$$

where

$$A_1 = -\alpha_5 \qquad B_2 = 1$$
$$A_o = \alpha_5 \qquad B_1 = \alpha_3 - 2 + \alpha_4\alpha_7 \tag{4-17}$$
$$B_o = 1 - \alpha_3$$

Now let

$$Z = e^{j\omega T_C} = e^{j\theta} \tag{4-18}$$

$$\frac{V_{BP}}{V_{in}} = \frac{A_1 \cos\theta + A_o + jA_1 \sin\theta}{B_2 \cos 2\theta + B_1 \cos\theta + B_o + j(B_2 \sin 2\theta + B_1 \sin\theta)} \tag{4-19}$$

Note that as $\omega \to 0$, the numerator of Eq. 4-19 approaches 0. This complex function can be evaluated in the usual manner to find the magnitude and phase angle as a function of the ω frequency.

B. MARTBP Computer-Aided Design Program

MARTBP is a computer-aided design program which simplifies the design of a second-order bandpass switched-capacitor filter. The program is called by the following commands:

```
OLD MARTBP
FORTRAN
RUN-10 MARTBP
```

After logging on, the menu of commands will appear as shown on the following page.

```
"A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,"
"ENTER THE NUMBER OF THE APPROPRIATE COMMAND."

"1.  ENTER ALL NEW PARAMETERS"
"2.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F"
"3.  FIND CAPACITOR VALUES AND TOTAL C"
"4.  PRINT MENU"
"5.  TERMINATE PROGRAM"
"6.  WARP FO, BW, AND PEAK GAIN."
"7.  PRINT CURRENT PARAMETER VALUES."
"8.  FIND MAGNITUDE AND PHASE AT A SERIES"
"    OF FREQUENCY POINTS."
"9.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON BW & FO."

"10.  CHANGE FO ONLY"
"11.  CHANGE BW ONLY"
"12.  CHANGE PEAK GAIN ONLY"
"13.  CHANGE FC ONLY"
"14.  CHANGE ALPHAC=C(2)/C(1) ONLY"
"15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4) ONLY"
"16.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER"
```

Initially, the designer would select Menu Command 1 to enter the bandpass parameters of f_o, BW, K (peak gain in dB), α_R, and f_c (the sampling frequency). MARTBP now has enough data to calculate all α values from Eqs. 4-6 through 4-9. Also, the complex frequency response can be determined by Eq. 4-19 and either Menu Command 2 or Menu Command 8.

Some program efficiency results if Menu Command 3 for finding capacitor values and the total capacitance is tried after entering all new parameters. Then Command 16 should be used to minimize the total capacitance by individually varying one of the "free" values $-\alpha_R$ and α_C. Next use Command 6 to warp the α values to compensate for a non-zero sampling period.*

Command 9 allows the designer to sweep the clock frequency and observe the effects on BW and f_o. Commands 10 through 15 permit the user to alter one design parameter without re-entering all data. Following such a change, the user can proceed with any other menu command as desired.

---

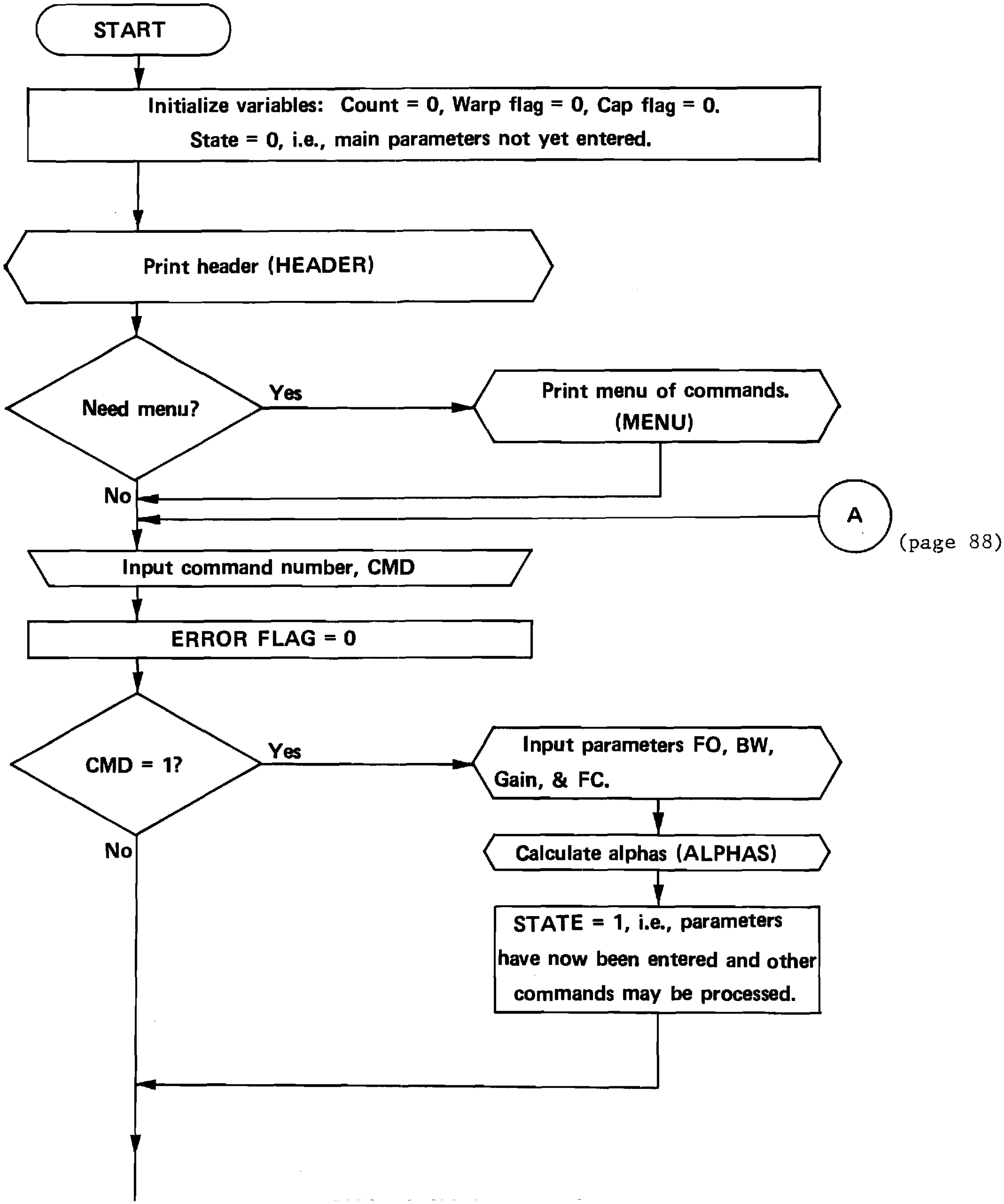
*For additional comments on the warping algorithm, refer to page 25 where the MARTLP program was discussed.

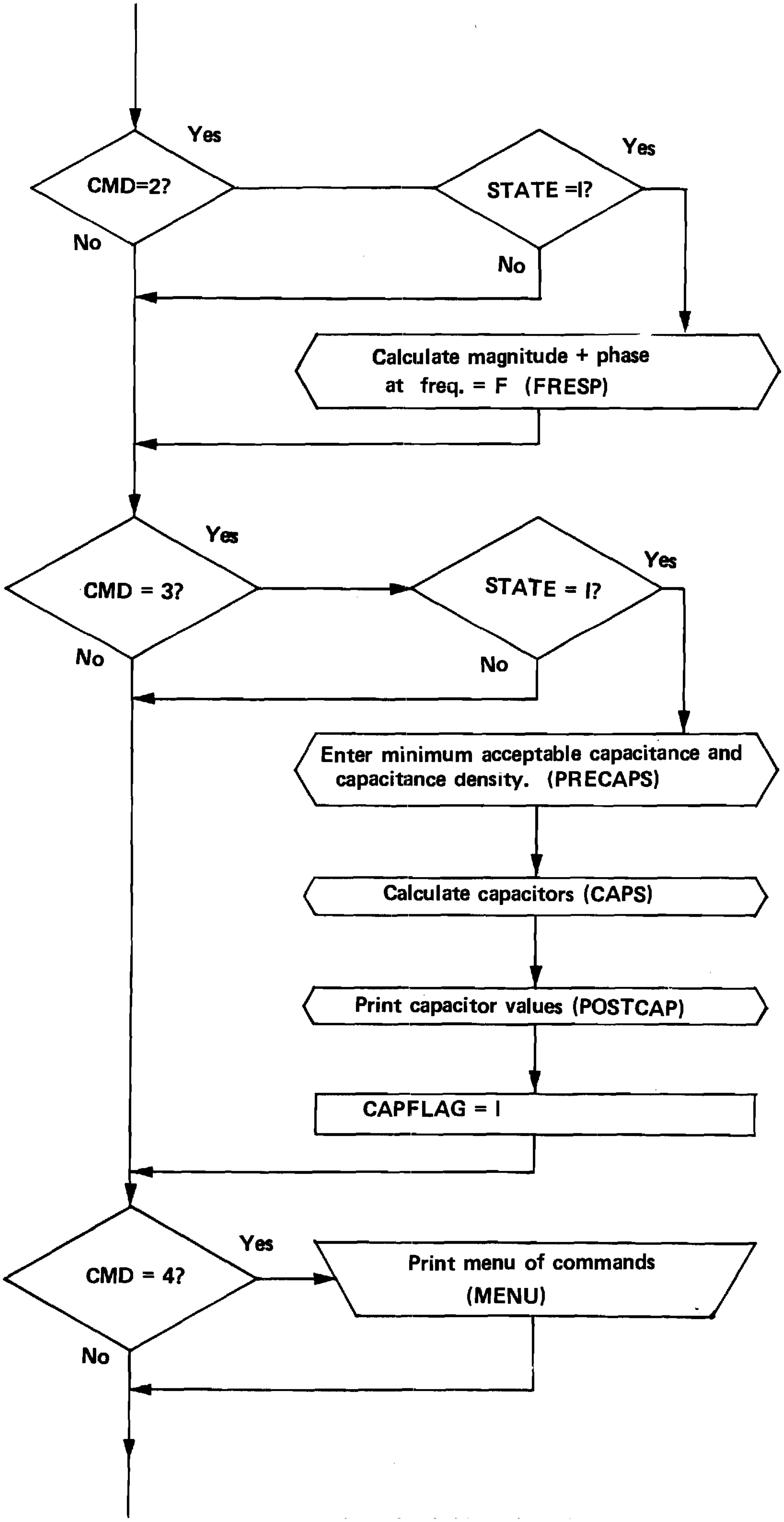
C. Flow Diagram for MARTBP

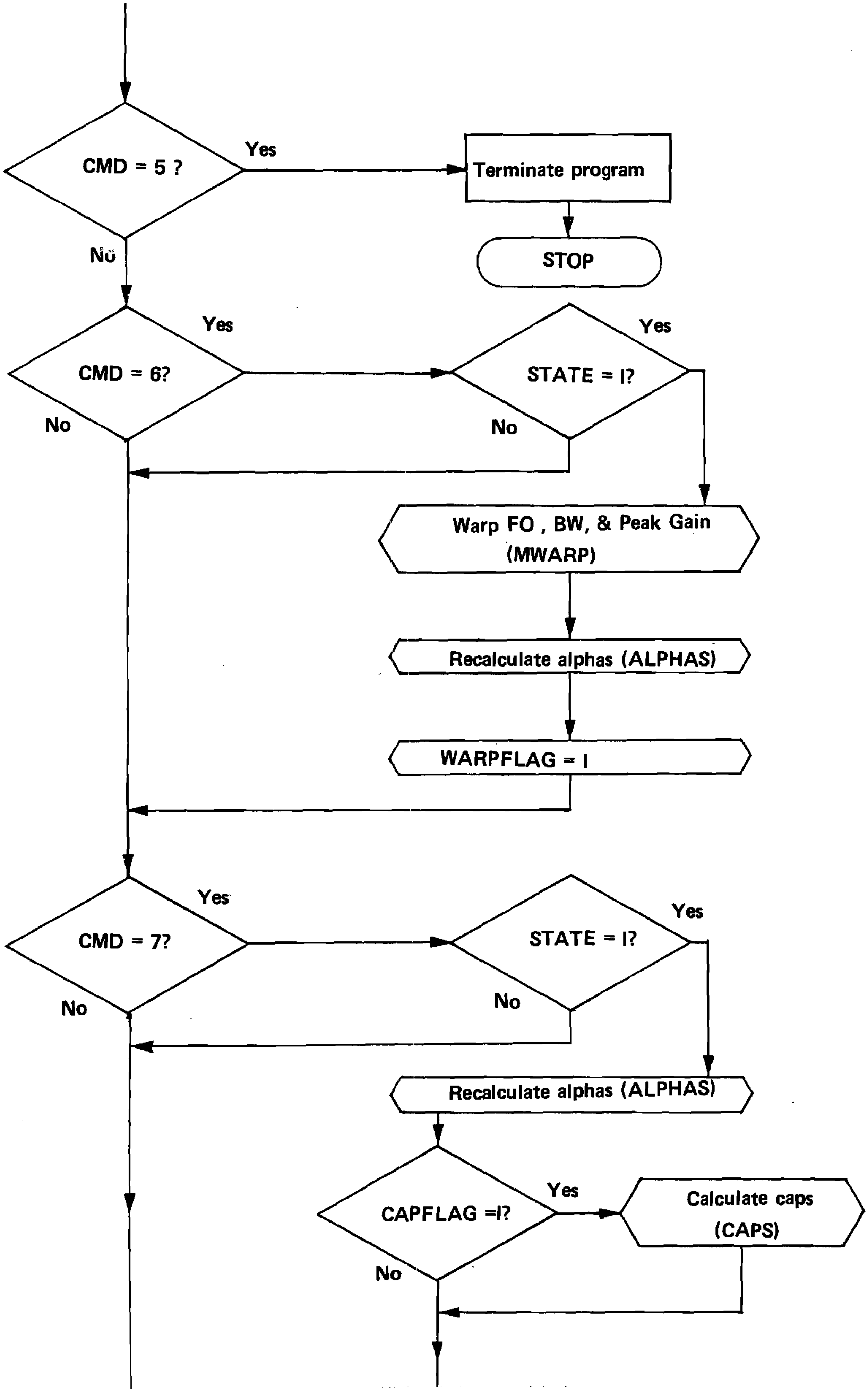
The flow chart showing program order and branching decisions follows. A complete FORTRAN listing for MARTBP is given in Appendix B.

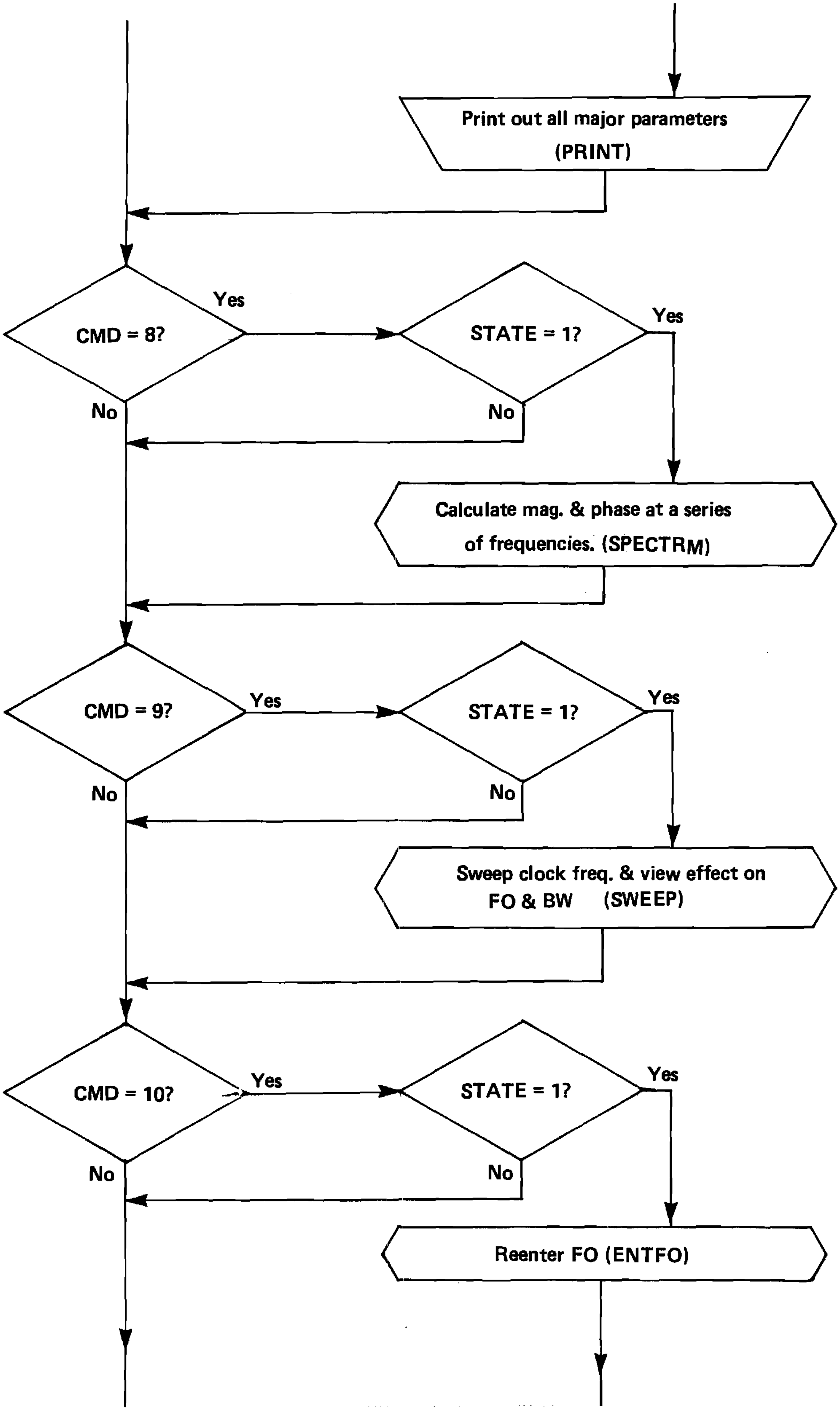
Program MARTBP

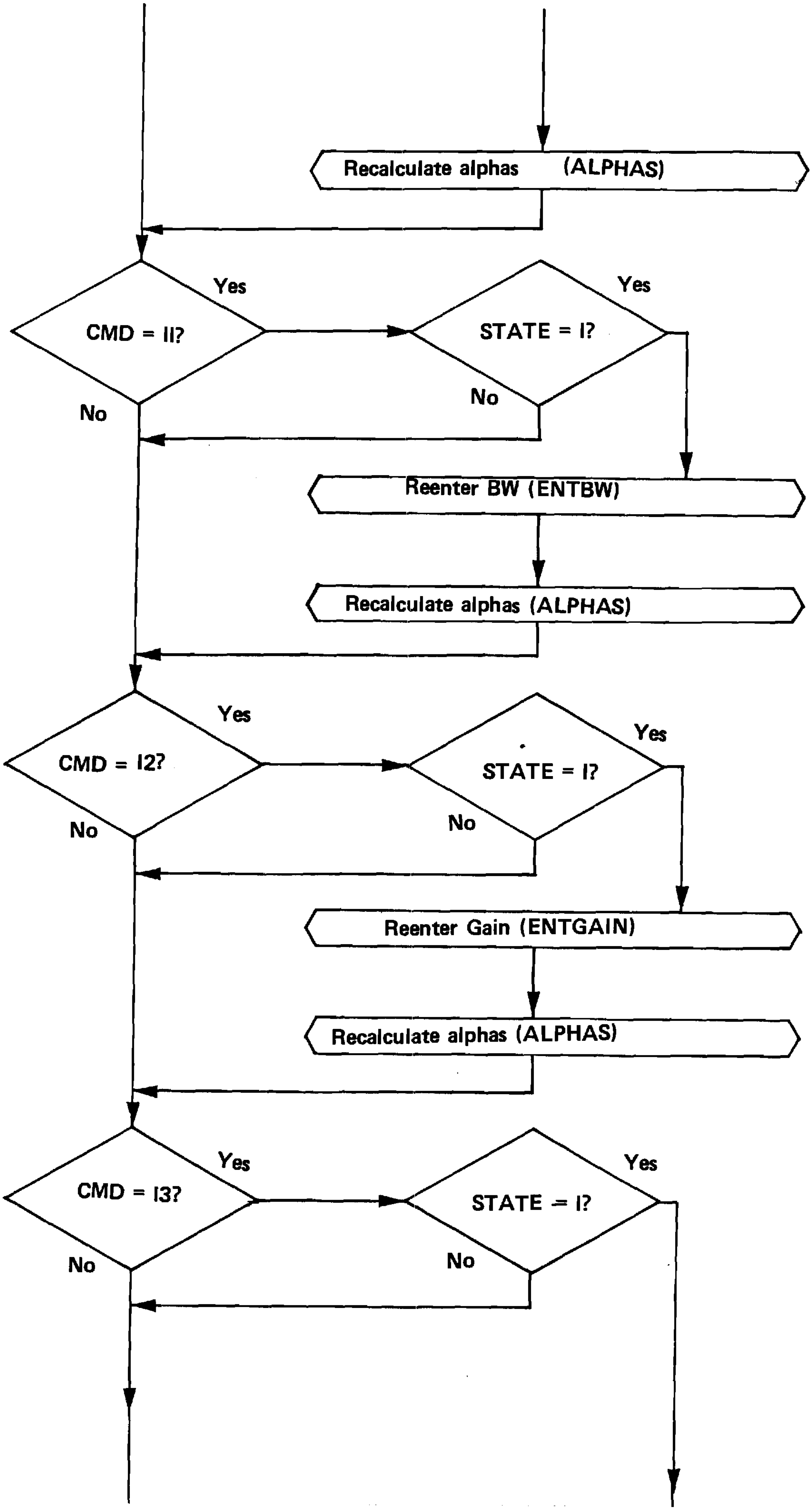
Main Program

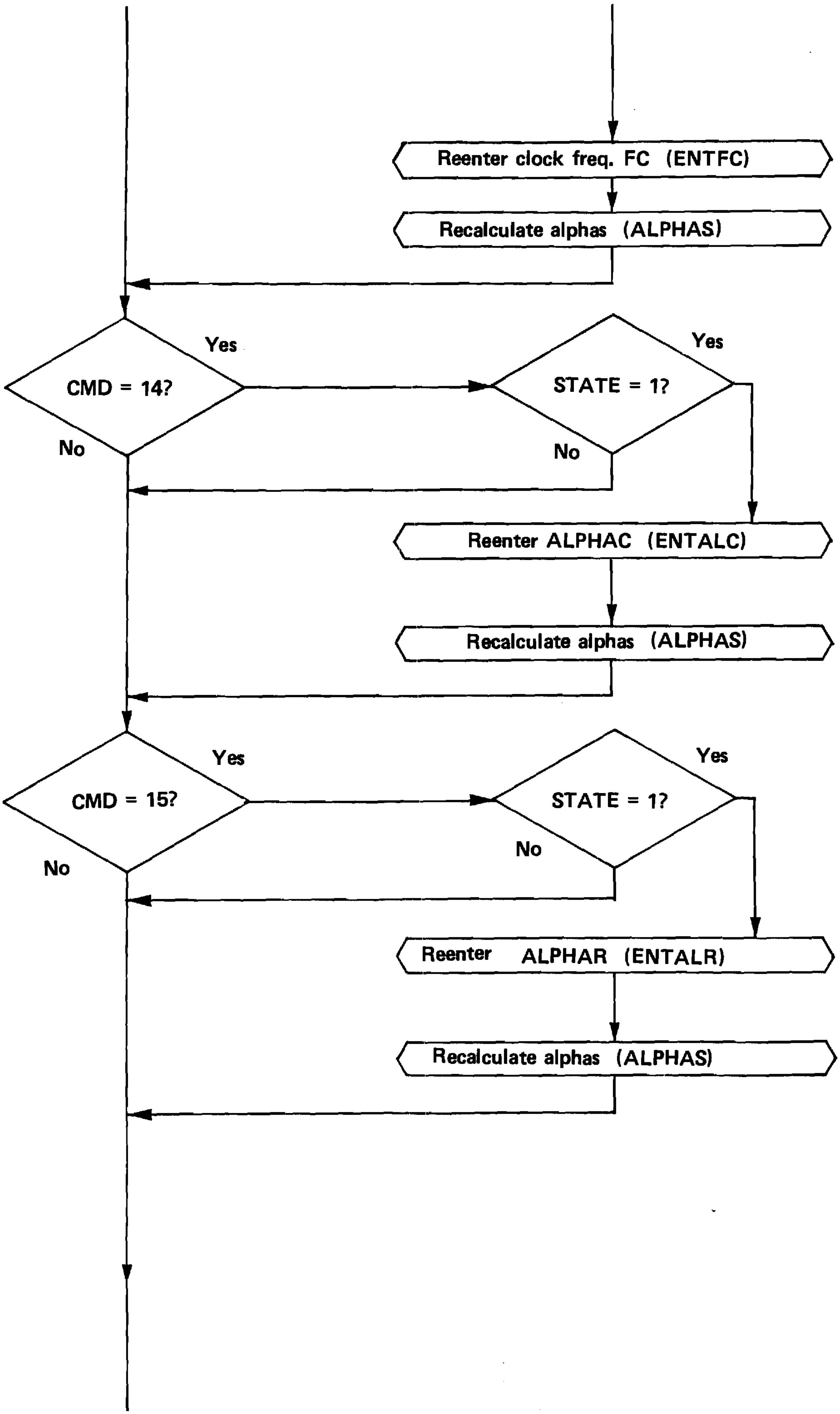


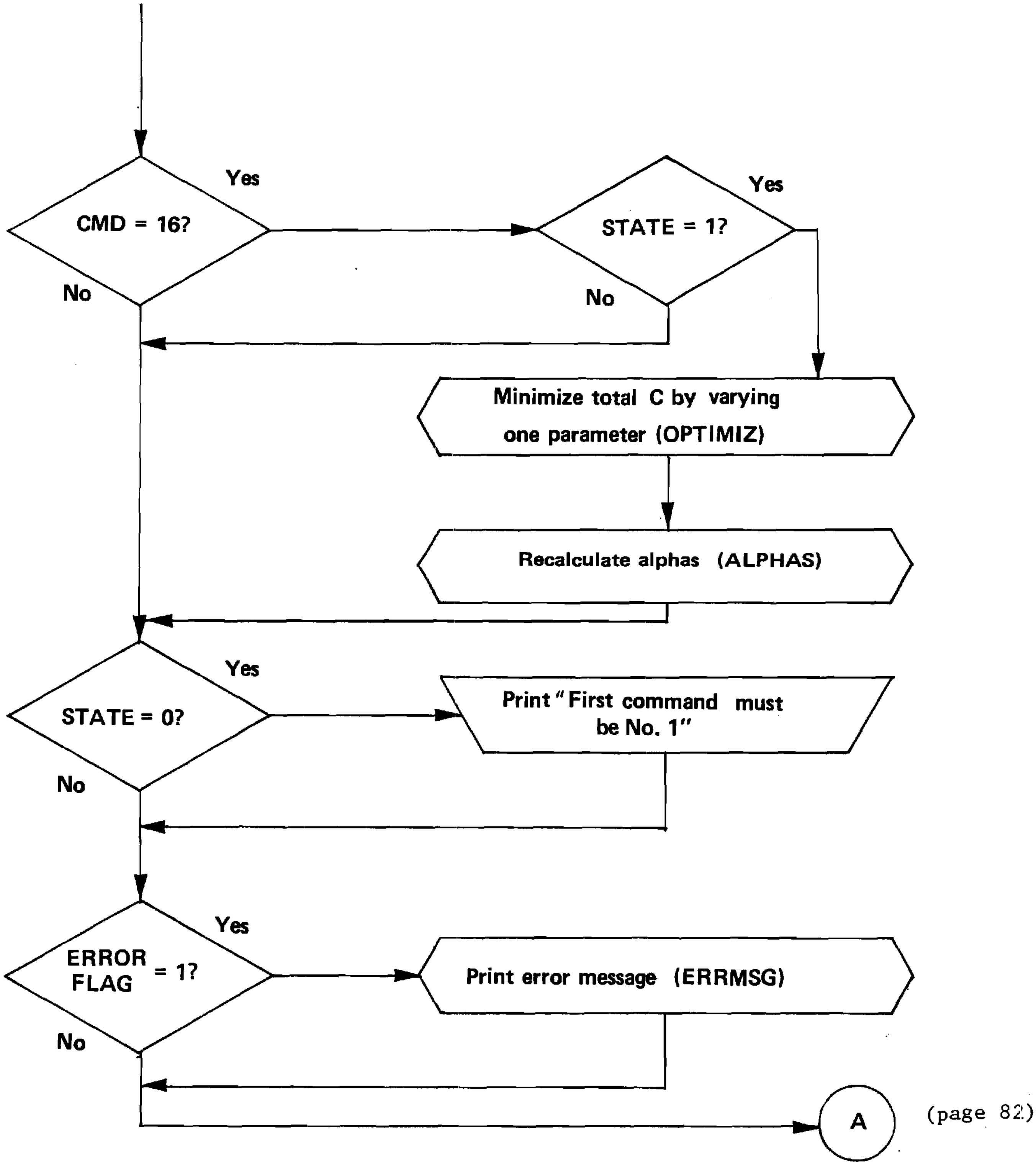

CMD=2?
Yes
No
STATE =I?
Yes
No
Calculate magnitude + phase at freq. = F (FRESP)
CMD = 3?
Yes
No
STATE = I?
Yes
No
Enter minimum acceptable capacitance and capacitance density. (PRECAPS)
Calculate capacitors (CAPS)
Print capacitor values (POSTCAP)
CAPFLAG = I
CMD = 4?
Yes
No
Print menu of commands (MENU)


CMD = 5 ?
Yes
Terminate program
No
STOP
CMD = 6?
Yes
No
STATE = I?
Yes
No
Warp FO , BW, & Peak Gain (MWARP)
Recalculate alphas (ALPHAS)
WARPFLAG = I
CMD = 7?
Yes
No
STATE = I?
Yes
No
Recalculate alphas (ALPHAS)
CAPFLAG =I?
Yes
No
Calculate caps (CAPS)


Print out all major parameters (PRINT)
CMD = 8?
Yes
No
STATE = 1?
Yes
No
Calculate mag. & phase at a series of frequencies. (SPECTRM)
CMD = 9?
Yes
No
STATE = 1?
Yes
No
Sweep clock freq. & view effect on FO & BW (SWEEP)
CMD = 10?
Yes
No
STATE = 1?
Yes
No
Reenter FO (ENTFO)

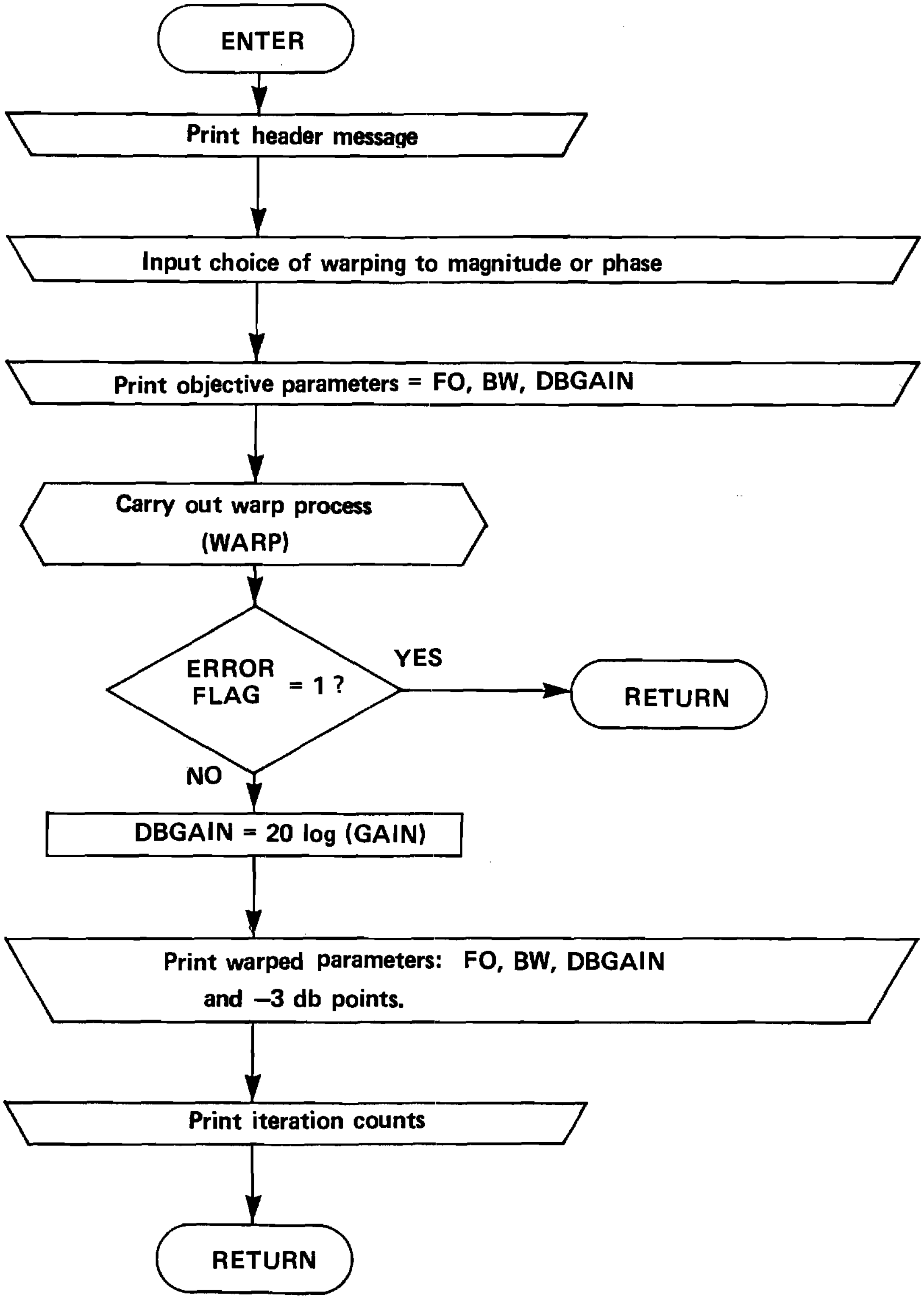

Recalculate alphas (ALPHAS)
CMD = 11?
Yes
No
STATE = 1?
Yes
No
Reenter BW (ENTBW)
Recalculate alphas (ALPHAS)
CMD = 12?
Yes
No
STATE = 1?
Yes
No
Reenter Gain (ENTGAIN)
Recalculate alphas (ALPHAS)
CMD = 13?
Yes
No
STATE = 1?
Yes
No


Reenter clock freq. FC (ENTFC)
Recalculate alphas (ALPHAS)
CMD = 14?
Yes
No
STATE = 1?
Yes
No
Reenter ALPHAC (ENTALC)
Recalculate alphas (ALPHAS)
CMD = 15?
Yes
No
STATE = 1?
Yes
No
Reenter ALPHAR (ENTALR)
Recalculate alphas (ALPHAS)

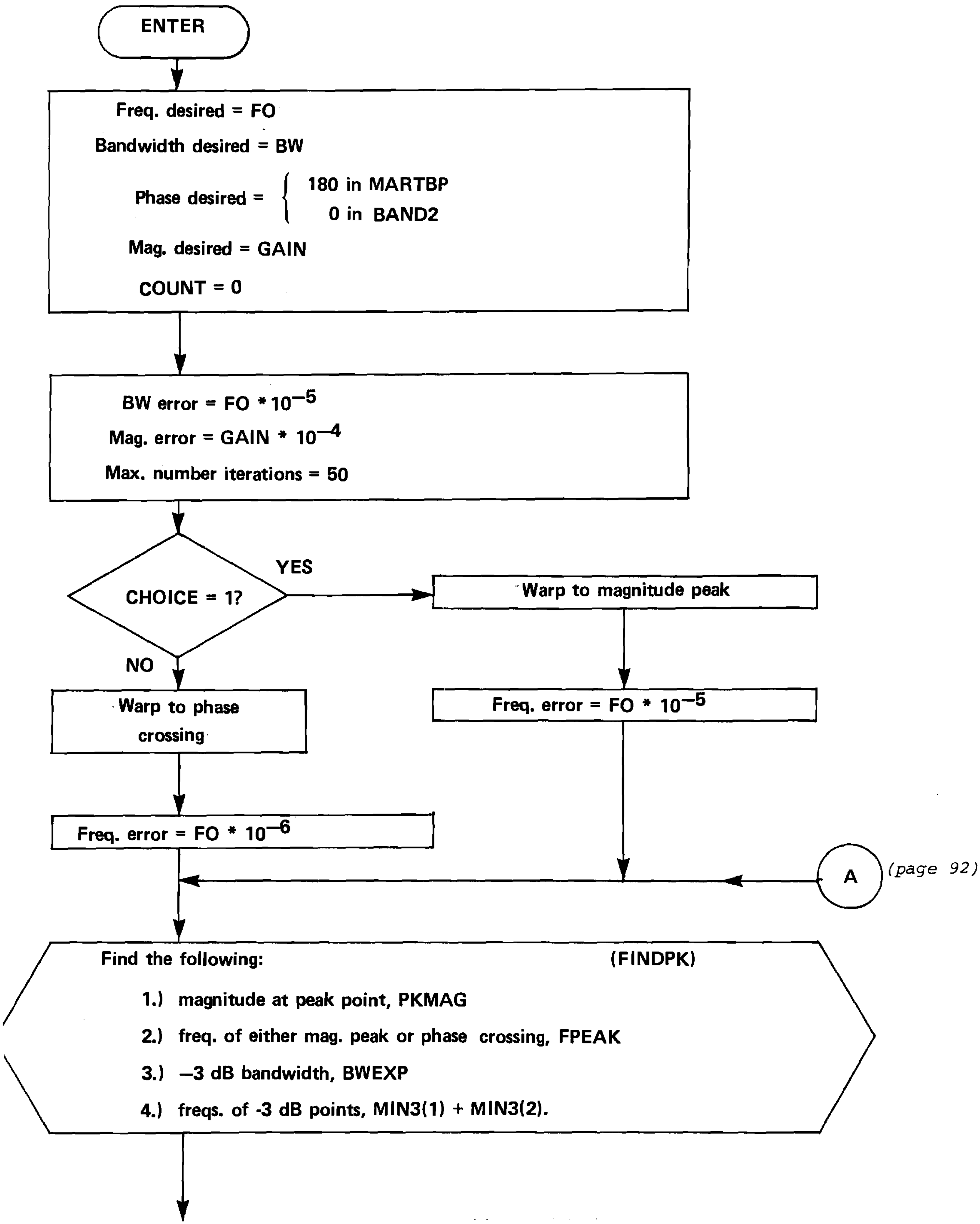

CMD = 16?
Yes
No
STATE = 1?
Yes
No
Minimize total C by varying one parameter (OPTIMIZ)
Recalculate alphas (ALPHAS)
STATE = 0?
Yes
No
Print "First command must be No. 1"
ERROR FLAG = 1?
Yes
No
Print error message (ERRMSG)
A
(page 82)

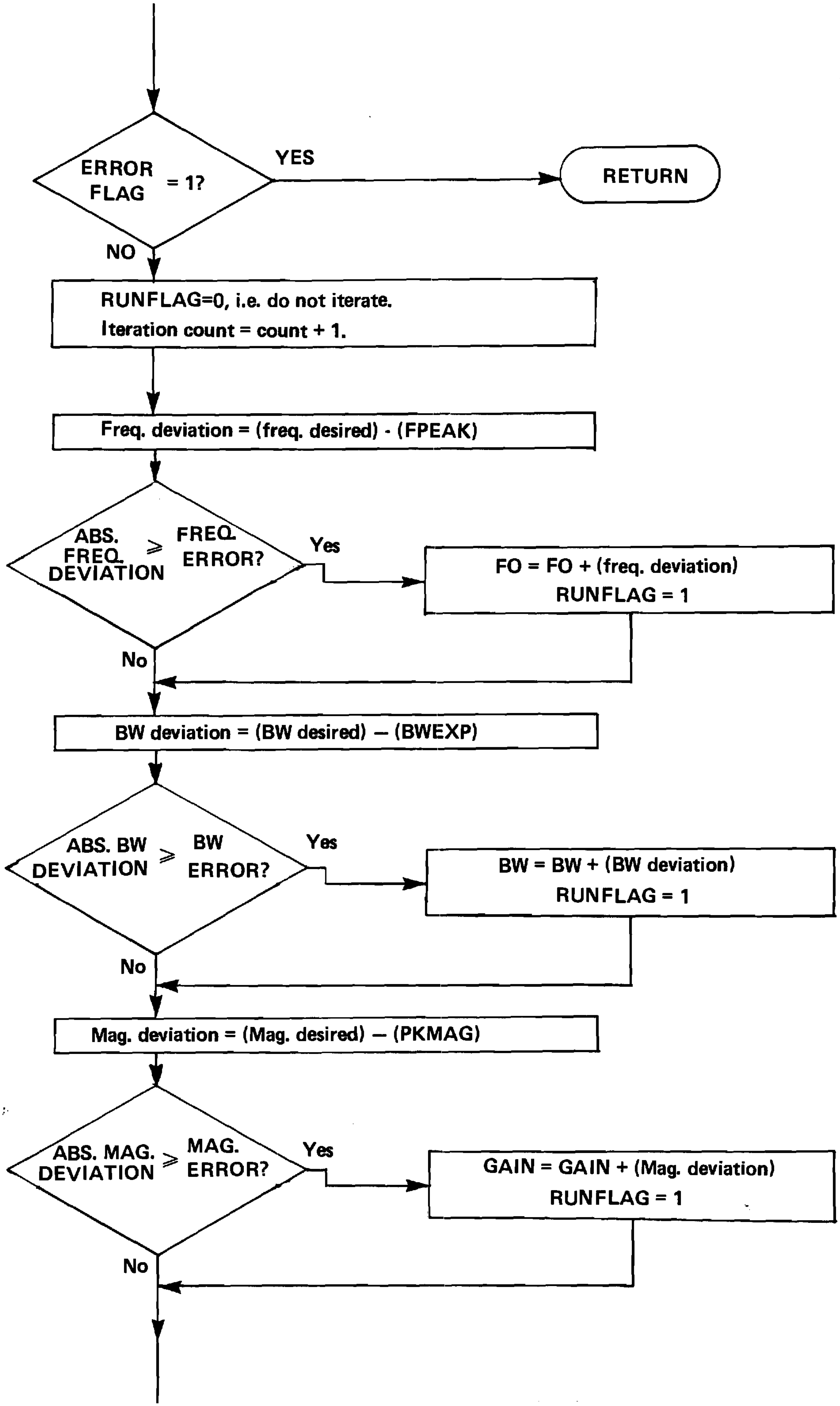
Warping routines for Programs MARTBP and BAND2

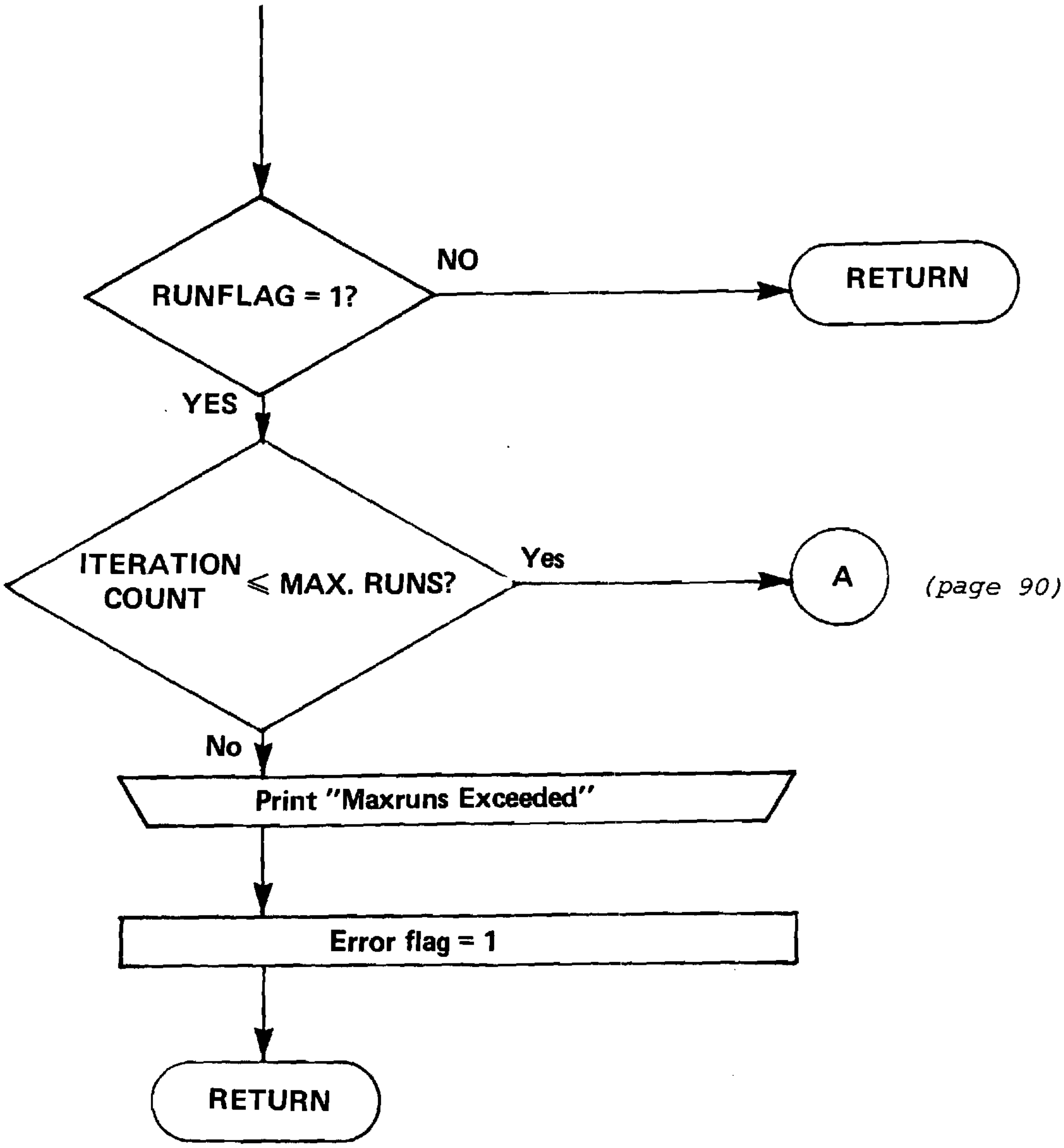
Subroutine MWARP

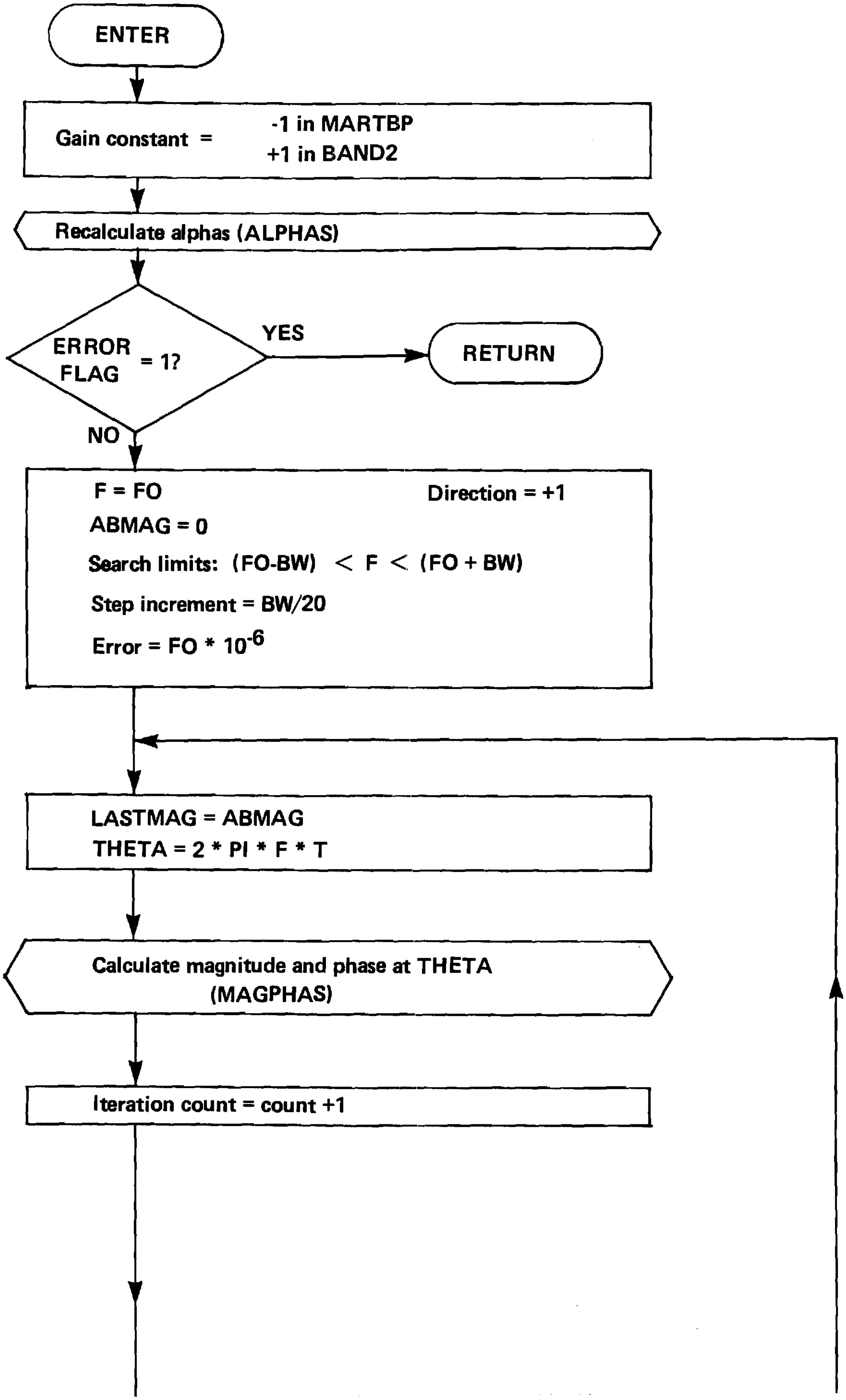


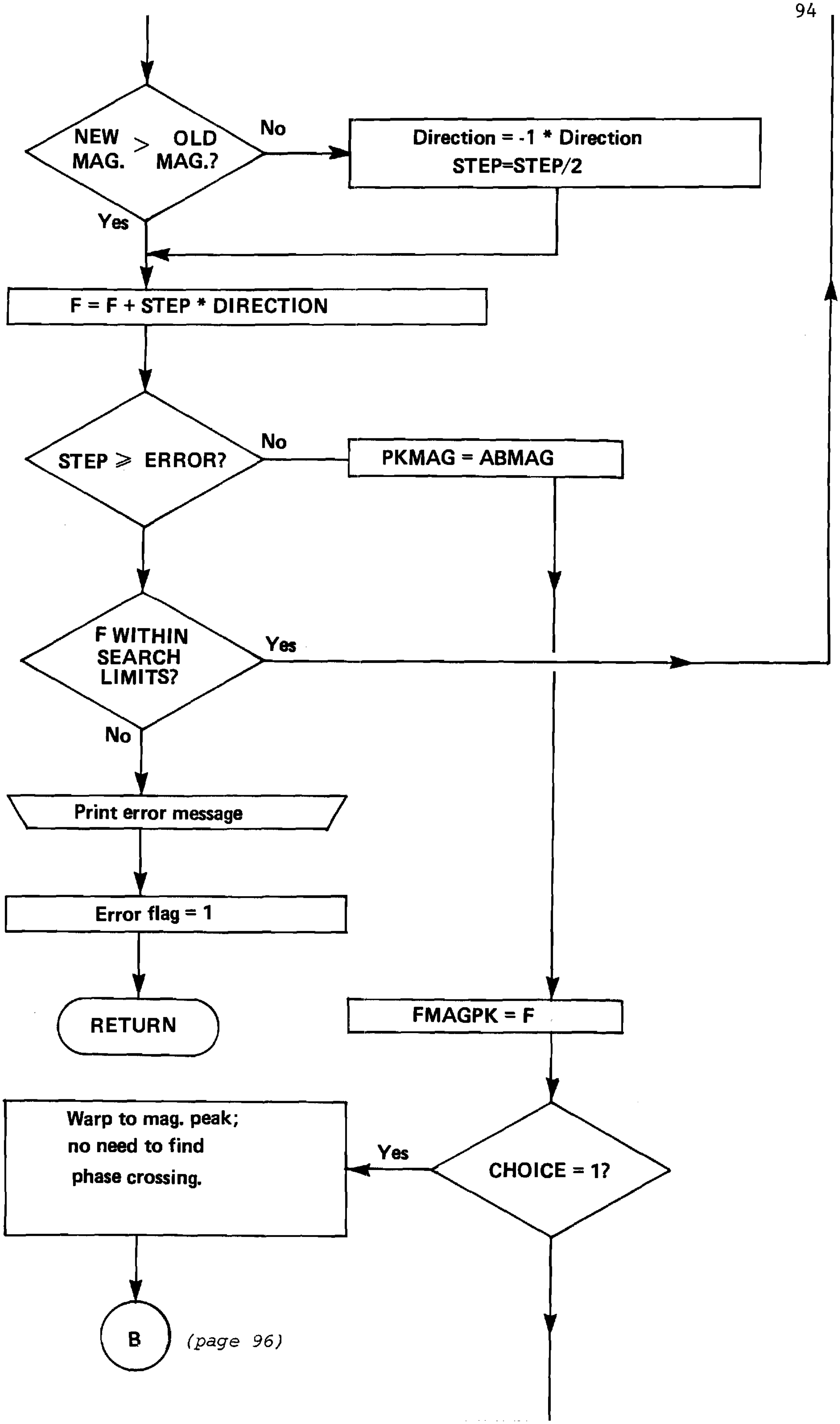
Subroutine WARP



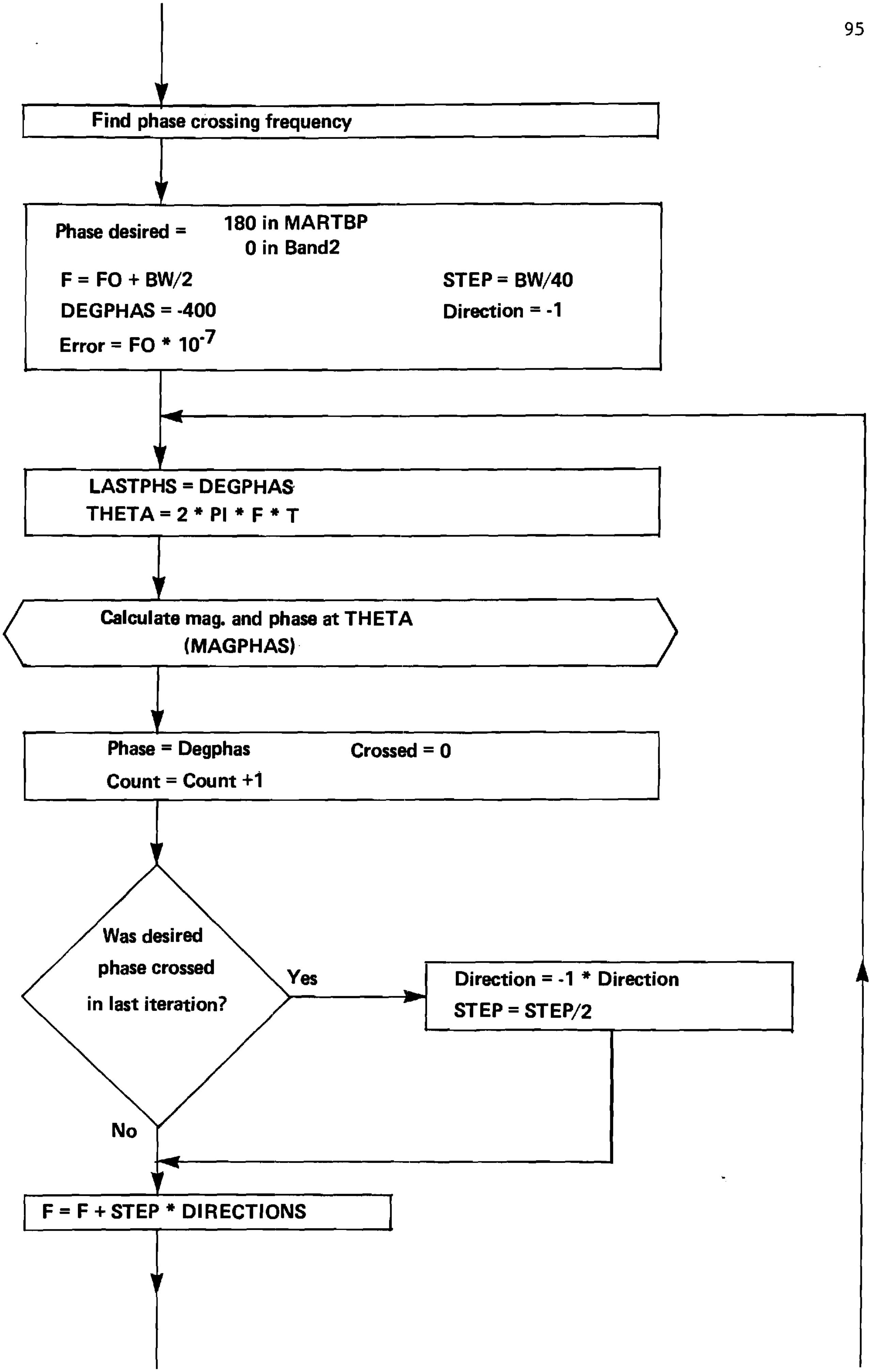

ERROR FLAG = 1?
YES
RETURN
NO
RUNFLAG=0, i.e. do not iterate.
Iteration count = count + 1.
Freq. deviation = (freq. desired) - (FPEAK)
ABS. FREQ. DEVIATION ⩾ FREQ. ERROR?
Yes
FO = FO + (freq. deviation)
RUNFLAG = 1
No
BW deviation = (BW desired) — (BWEXP)
ABS. BW DEVIATION ⩾ BW ERROR?
Yes
BW = BW + (BW deviation)
RUNFLAG = 1
No
Mag. deviation = (Mag. desired) — (PKMAG)
ABS. MAG. DEVIATION ⩾ MAG. ERROR?
Yes
GAIN = GAIN + (Mag. deviation)
RUNFLAG = 1
No

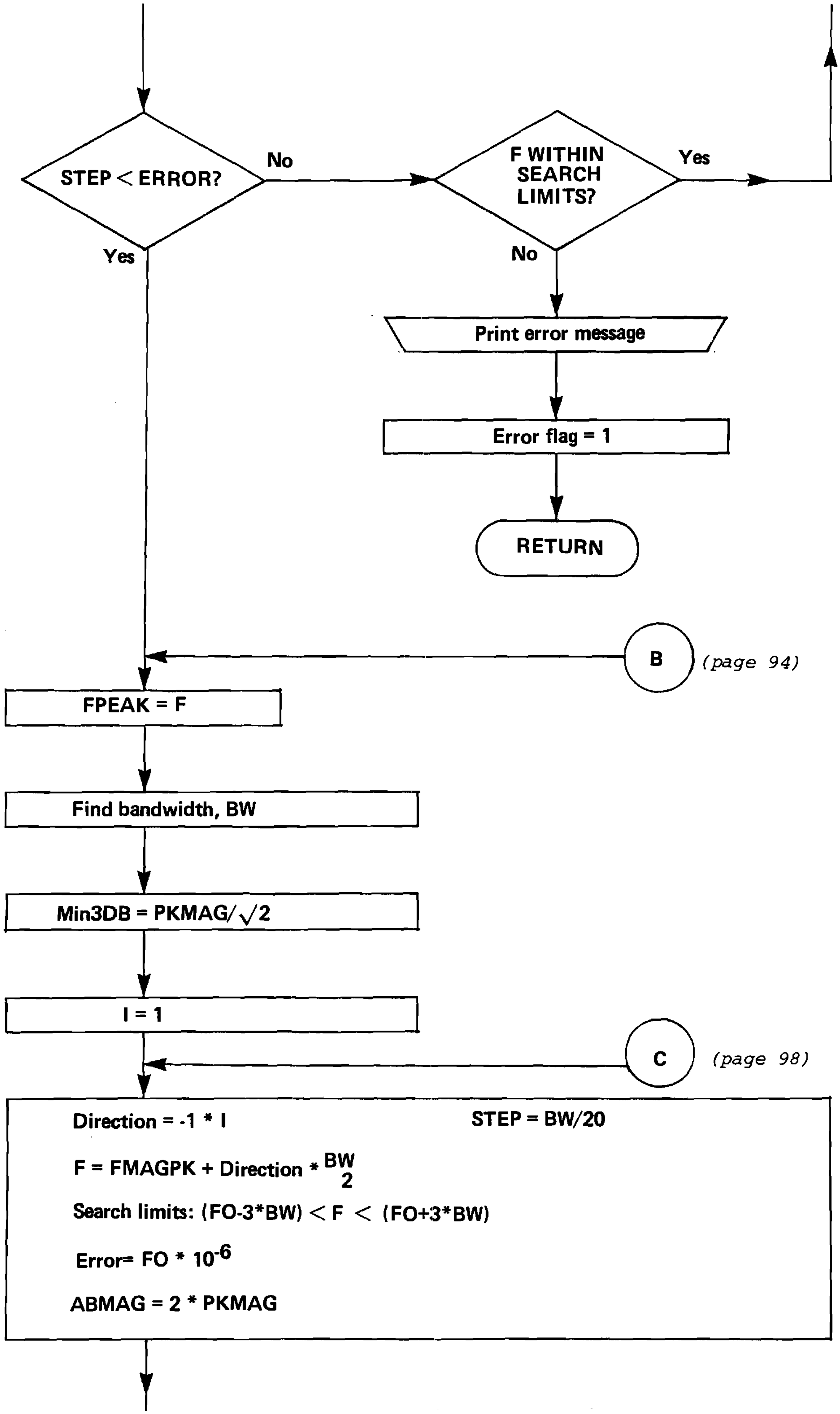

RUNFLAG = 1?
NO
RETURN
YES
ITERATION COUNT ≤ MAX. RUNS?
Yes
A
(page 90)
No
Print "Maxruns Exceeded"
Error flag = 1
RETURN

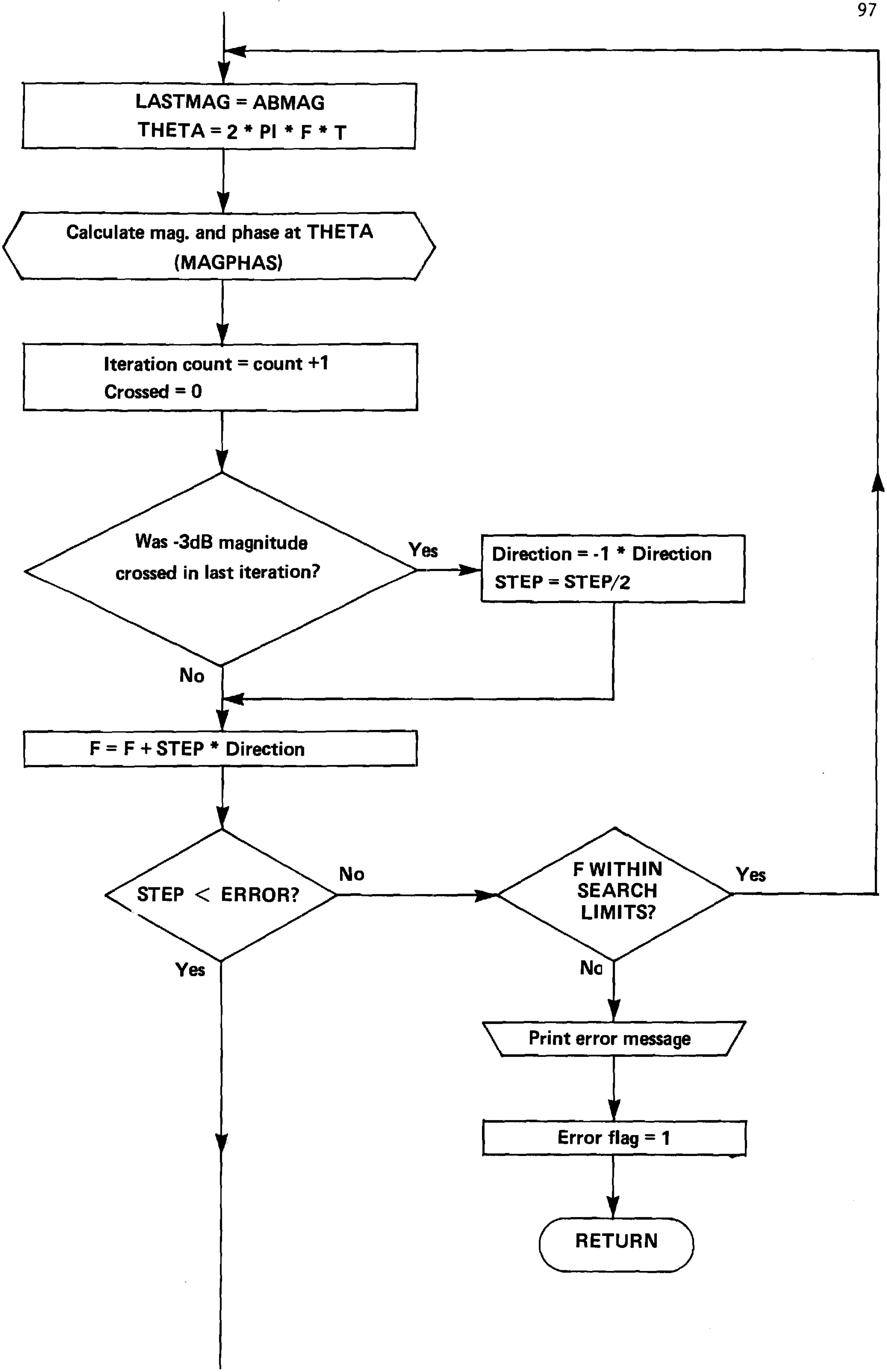

ENTER
Gain constant = -1 in MARTBP
+1 in BAND2
Recalculate alphas (ALPHAS)
ERROR FLAG = 1?
YES
RETURN
NO
F = FO
Direction = +1
ABMAG = 0
Search limits: (FO-BW) < F < (FO + BW)
Step increment = BW/20
Error = FO * 10-6
LASTMAG = ABMAG
THETA = 2 * PI * F * T
Calculate magnitude and phase at THETA (MAGPHAS)
Iteration count = count +1

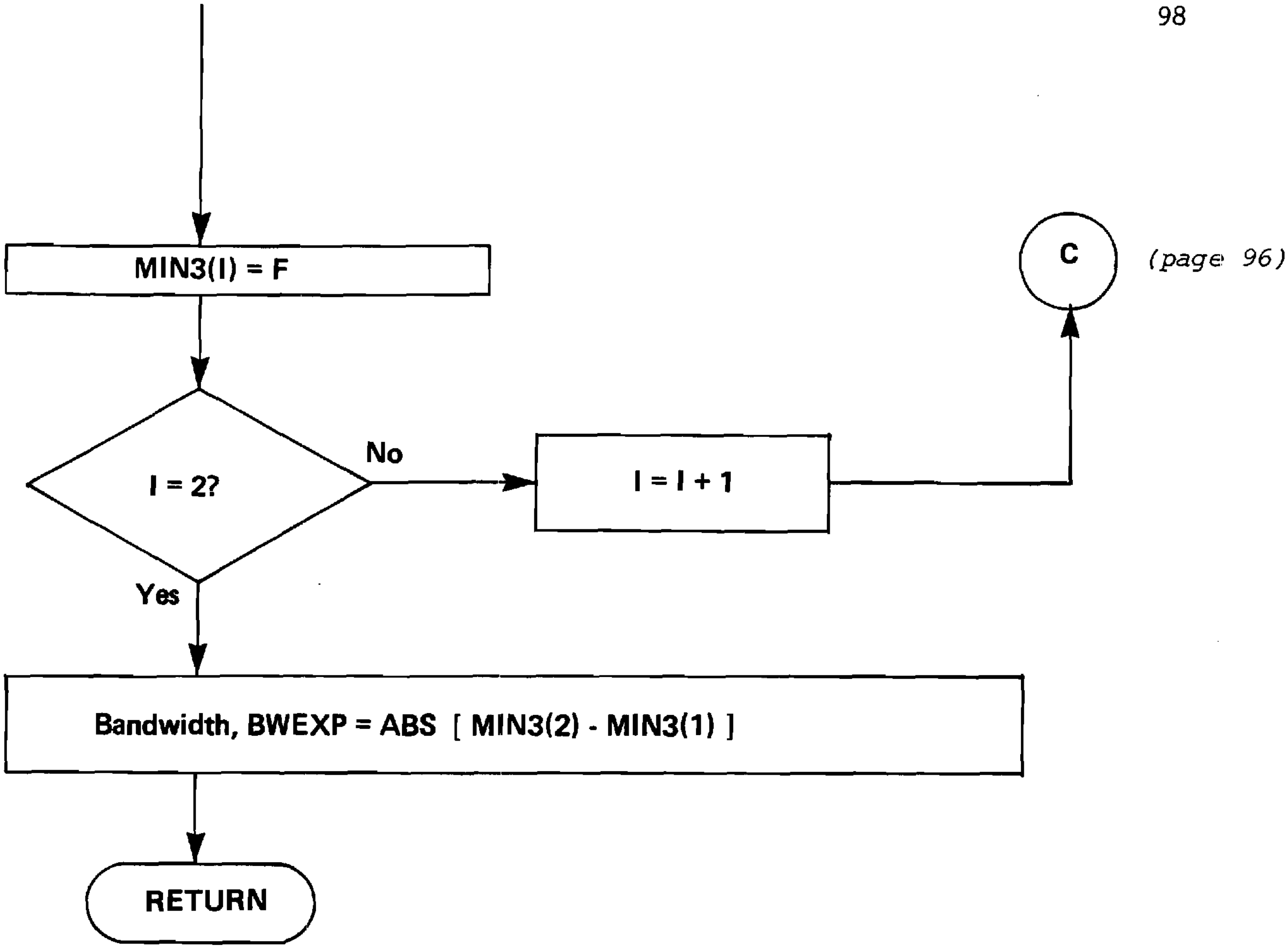

NEW MAG. > OLD MAG.?
No
Direction = -1 * Direction
STEP=STEP/2
Yes
F = F + STEP * DIRECTION
STEP ⩾ ERROR?
No
PKMAG = ABMAG
F WITHIN SEARCH LIMITS?
Yes
No
Print error message
Error flag = 1
RETURN
FMAGPK = F
Warp to mag. peak; no need to find phase crossing.
Yes
CHOICE = 1?
B

(page 96)


Find phase crossing frequency
Phase desired = 180 in MARTBP
0 in Band2
F = FO + BW/2
STEP = BW/40
DEGPHAS = -400
Direction = -1
Error = FO * 10^-7
LASTPHS = DEGPHAS
THETA = 2 * PI * F * T
Calculate mag. and phase at THETA
(MAGPHAS)
Phase = Degphas
Crossed = 0
Count = Count +1
Was desired
phase crossed
in last iteration?
Yes
Direction = -1 * Direction
STEP = STEP/2
No
F = F + STEP * DIRECTIONS


STEP < ERROR?
No
Yes
F WITHIN SEARCH LIMITS?
Yes
No
Print error message
Error flag = 1
RETURN
B
(page 94)
FPEAK = F
Find bandwidth, BW
Min3DB = PKMAG/√2
I = 1
C
(page 98)
Direction = -1 * I
STEP = BW/20
F = FMAGPK + Direction * BW/2
Search limits: (FO-3*BW) < F < (FO+3*BW)
Error= FO * 10-6
ABMAG = 2 * PKMAG


LASTMAG = ABMAG
THETA = 2 * PI * F * T
Calculate mag. and phase at THETA (MAGPHAS)
Iteration count = count +1
Crossed = 0
Was -3dB magnitude crossed in last iteration?
Yes
Direction = -1 * Direction
STEP = STEP/2
No
F = F + STEP * Direction
STEP < ERROR?
No
F WITHIN SEARCH LIMITS?
Yes
Yes
No
Print error message
Error flag = 1
RETURN


MIN3(I) = F
I = 2?
No
I = I + 1
C
(page 96)
Yes
Bandwidth, BWEXP = ABS [MIN3(2) - MIN3(1)]
RETURN

D. Sample Bandpass Design Using MARTBP

```
*OLD MARTBP
*FORTRAN
*RUN-10 MARTBP
SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR
MARTIN STATE-VARIABLE BANDPASS FILTER.
ALGORITHM BY J.A. CONNELLY, FORTRAN VERSION BY
W.N. GOOLSBY.  CURRENT REVISION 6/5/81.

NEED MENU? (0=NO, 1=YES)
=1
A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,
ENTER THE NUMBER OF THE APPROPRIATE COMMAND.


1.  ENTER ALL NEW PARAMETERS
2.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F
3.  FIND CAPACITOR VALUES AND TOTAL C
4.  PRINT MENU
5.  TERMINATE PROGRAM
6.  WARP FO, BW, AND PEAK GAIN.
7.  PRINT CURRENT PARAMETER VALUES.
8.  FIND MAGNITUDE AND PHASE AT A SERIES
    OF FREQUENCY POINTS.
9.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON BW & FO.

10.  CHANGE FO ONLY
11.  CHANGE BW ONLY
12.  CHANGE PEAK GAIN ONLY
13.  CHANGE FC ONLY
14.  CHANGE ALPHAC=C(2)/C(1) ONLY
15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4) ONLY
16.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER


ENTER COMMAND NUMBER (#4 FOR MENU)
=1
ENTER POLE FREQUENCY, FO (HERTZ)
=3000
ENTER POLE BANDWIDTH, BW (HERTZ)
=500
ENTER PEAK GAIN (DECIBELS)
=0
ENTER ALPHAC=C(2)/C(1)
=1
ENTER ALPHAR=ALPHA(7)/ALPHA(4)
=1
ENTER SAMPLING FREQUENCY, FC (HERTZ)
=100000
```

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=3
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF

C(1)=   31.830989      PF
C(2)=   31.830989      PF
C(3)=   1.0000000      PF
C(4)=   6.0000000      PF
C(5)=   1.0000000      PF
C(7)=   6.0000000      PF

TOTAL CAPACITANCE USED=   77.661977      PF
AT  0.2000     PF/SQ, AREA USED =   388.310      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=16
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  ALPHAR=ALPHA(7)/ALPHA(4)
=1

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAC
FOR PARAMETER=      0.17000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   46.262257      PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=16
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  ALPHAR=ALPHA(7)/ALPHA(4)
=2

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.1,10,.01
PARAMETER NAME:
ALPHAR
FOR PARAMETER=      0.17000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   44.189983      PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.
```

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=3
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF

C(1)=   31.830989      PF
C(2)=   5.4112681      PF
C(3)=   1.0000000      PF
C(4)=   2.4738634      PF
C(5)=   1.0000000      PF
C(7)=   2.4738634      PF

TOTAL CAPACITANCE USED=   44.189983      PF
AT  0.2000     PF/SQ, AREA USED =   220.950      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=8
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	-0.36813560E-01	0.99577065	182.14350
2996.0000	-0.32359233E-01	0.99628144	181.91746
2997.0000	-0.28037294E-01	0.99677729	181.69127
2998.0000	-0.23848004E-01	0.99725816	181.46493
2999.0000	-0.19791612E-01	0.99772400	181.23846
3000.0000	-0.15868355E-01	0.99817476	181.01186
3001.0000	-0.12078458E-01	0.99861038	180.78514
3002.0000	-0.84221352E-02	0.99903084	180.55830
3003.0000	-0.48995877E-02	0.99943607	180.33134
3004.0000	-0.15110051E-02	0.99982605	180.10429
3005.0000	0.17434353E-02	1.0002007	179.87714

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=6
FINDS PREWARPED VALUES FOR FO, BW, AND PEAK GAIN.
YOU MAY CHOOSE TO WARP FO TO MAGNITUDE PEAK
OR TO 180 DEGREE PHASE CROSSING.
ENTER NUMBER OF DESIRED COMMAND.
     1.  MAGNITUDE PEAK
     2.  180 DEGREE PHASE CROSSING
=1
DESIRED BANDPASS CENTER=   3000.0000    HZ
DESIRED BANDWIDTH=   500.00000    HZ
DESIRED PEAK GAIN=  0.              DECIBELS
```

```
PREWARPED VARIABLES ARE:  FO=   2972.1501    HZ
BW=   492.19343    HZ
PEAK GAIN= -0.38358722E-01  DECIBELS
THE -3DB POINTS WERE FOUND AT:
F1=   2760.4804       AND F2=   3260.4876      HERTZ

WARP ITERATION COUNT=   4 CYCLES
FINDPK ITERATION COUNT=   334  CYCLES.

ENTER COMMAND NUMBER (#4 FOR MENU)
=3
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF

C(1)=   32.478971      PF
C(2)=   5.5214250      PF
C(3)=   1.0044260      PF
C(4)=   2.5007905      PF
C(5)=   1.0000000      PF
C(7)=   2.5007905      PF

TOTAL CAPACITANCE USED=   45.006403      PF
AT  0.2000     PF/SQ, AREA USED =   225.032      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=8
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	-0.14823214E-02	0.99982936	175.71857
2996.0000	-0.85247891E-03	0.99990186	175.48729
2997.0000	-0.36207358E-03	0.99995832	175.25606
2998.0000	-0.11012043E-04	0.99999873	175.02488
2999.0000	0.20081231E-03	1.0000231	174.79377
3000.0000	0.27351926E-03	1.0000315	174.56273
3001.0000	0.20724176E-03	1.0000239	174.33176
3002.0000	0.21258404E-05	1.0000002	174.10088
3003.0000	-0.34166937E-03	0.99996066	173.87009
3004.0000	-0.82397174E-03	0.99990514	173.63940
3005.0000	-0.14445962E-02	0.99983370	173.40881

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=5
```

V. NOTCH FILTER

A. Design Equations

Fig. 5.1 shows a second-order notch filter with the following desirable features:

1. The noninverting input terminals of all op amps are grounded for reduced sensitivity to input and output impedances.
2. High Q rejection is possible.
3. All critical design parameters are independently adjustable.

The transfer function to the notch output terminal, V_N, is

$$\frac{V_N}{V_{in}} = -\alpha_3 \left[\frac{s^2 + \dfrac{C_4 s}{\alpha_3 R_7 C_1 C_2} + \dfrac{1}{\alpha_3 R_8 R_7 C_1 C_2}}{s^2 + \dfrac{s}{R_5 C_1} + \dfrac{1}{R_7 R_6 C_1 C_2}} \right] \tag{5-1}$$

$$= -\alpha_3 \left[\frac{s^2 + B_n s + \omega_n^2}{s^2 + B_d s + \omega_d^2} \right] \tag{5-2}$$

Note that B_n, the radian bandwidth of the numerator, is separately and independently adjustable via C_4. The zero frequency of the numerator, ω_n, is individually controlled by R_8. Resistor R_6 influences only ω_d, the pole frequency of the denominator. Lastly R_5 individually controls B_d, the bandwidth of the denominator. At very low frequencies, the magnitude of the transfer function approaches R_6/R_8. At very high frequencies this magnitude approaches α_3. In order to produce symmetrical asymptotic responses at low and high frequencies, set

$$\alpha_3 = R_6/R_8 \tag{5-3}$$

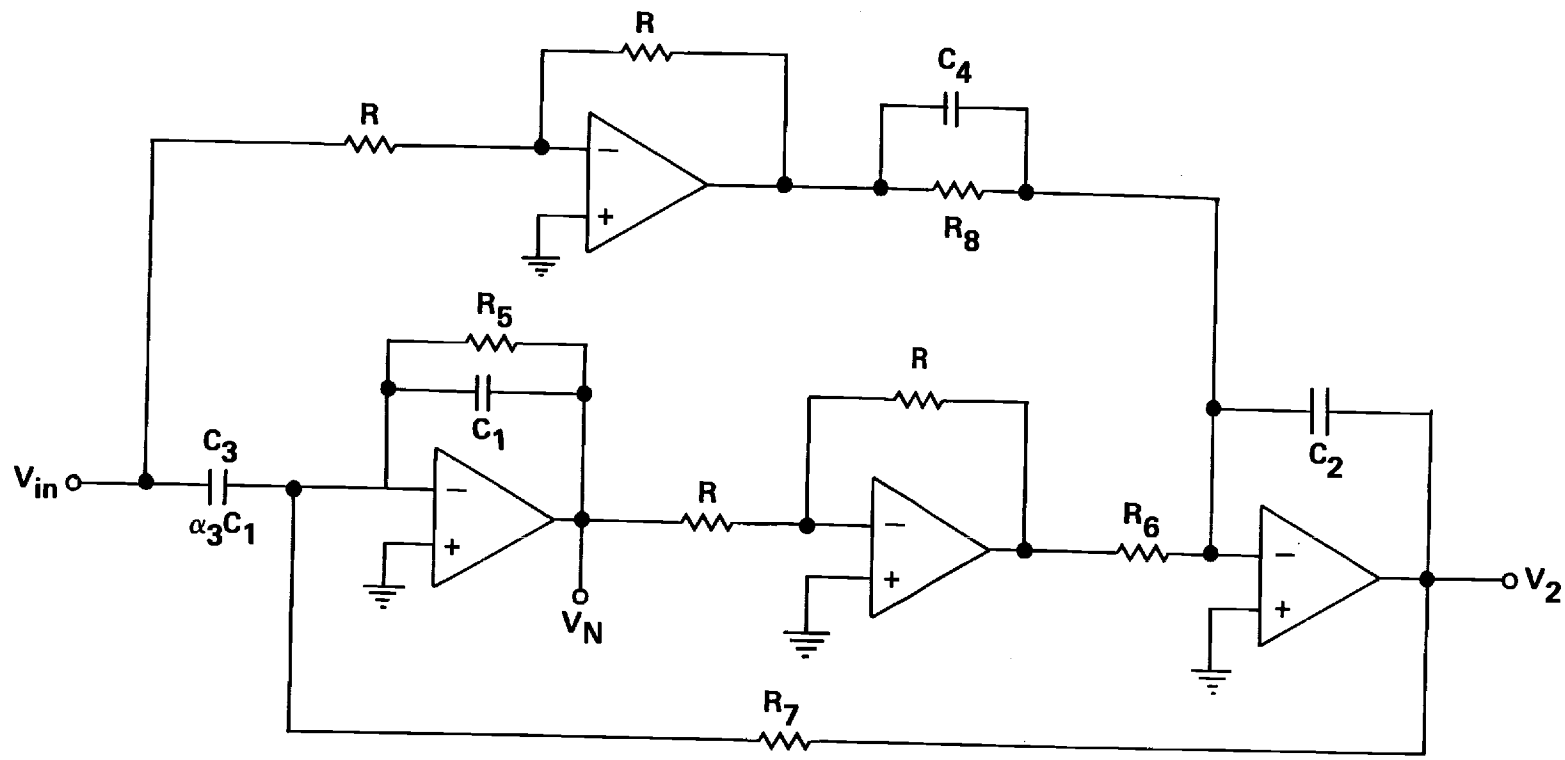


Fig. 5.1 Second-order analog notch filter.

The simplest notch filter design philosophy is to set $\omega_n = \omega_d = \omega_o$. This frequency then becomes the center frequency of the notch. Here the magnitude of Eq. 5-2 reduces to simply

$$\left| \frac{V_N}{V_{in}} \right|_{\omega=\omega_o} = \frac{\alpha_3 B_n}{B_d} = \frac{R_5 C_4}{R_7 C_2} \tag{5-4}$$

Thus, C_4 can be used to provide a finite notch depth at ω_o. Alternatively, maximum rejection occurs with C_4 omitted from the circuit. With $C_4 = 0$, the theoretical notch depth is $-\infty$, while the practical depth is limited by parasitic capacitances.

A switched-capacitor implementation of the analog notch filter is shown in Fig. 5.2. This SCF circuit is a modification of the configuration described by Gregorian [5]. Note, that this circuit uses only two op amps instead of the four required for the analog filter. The two simple analog inverters can be omitted by using the noninverting integrator SCF second stage.

Using the α substitution from Eq. 3-8, Eq. 5-1 simplifies to

$$\frac{V_N}{V_{in}} = -\alpha_3 \left[\frac{s^2 + \alpha_4 \alpha_7 s/\alpha_3 T_C + \alpha_7 \alpha_8/\alpha_3 T_C{}^2}{s^2 + \alpha_5 s/T_C + \alpha_6 \alpha_7/T_C{}^2} \right] \tag{5-5}$$

There are six α values in this equation and only four design parameters in Eq. 5-2. Therefore, choose $\alpha_3 = 1$ and define

$$\rho = \alpha_7/\alpha_6 \tag{5-6}$$

Now corresponding terms between Eqs. 5-2 and 5-5 can be compared to evaluate the α values

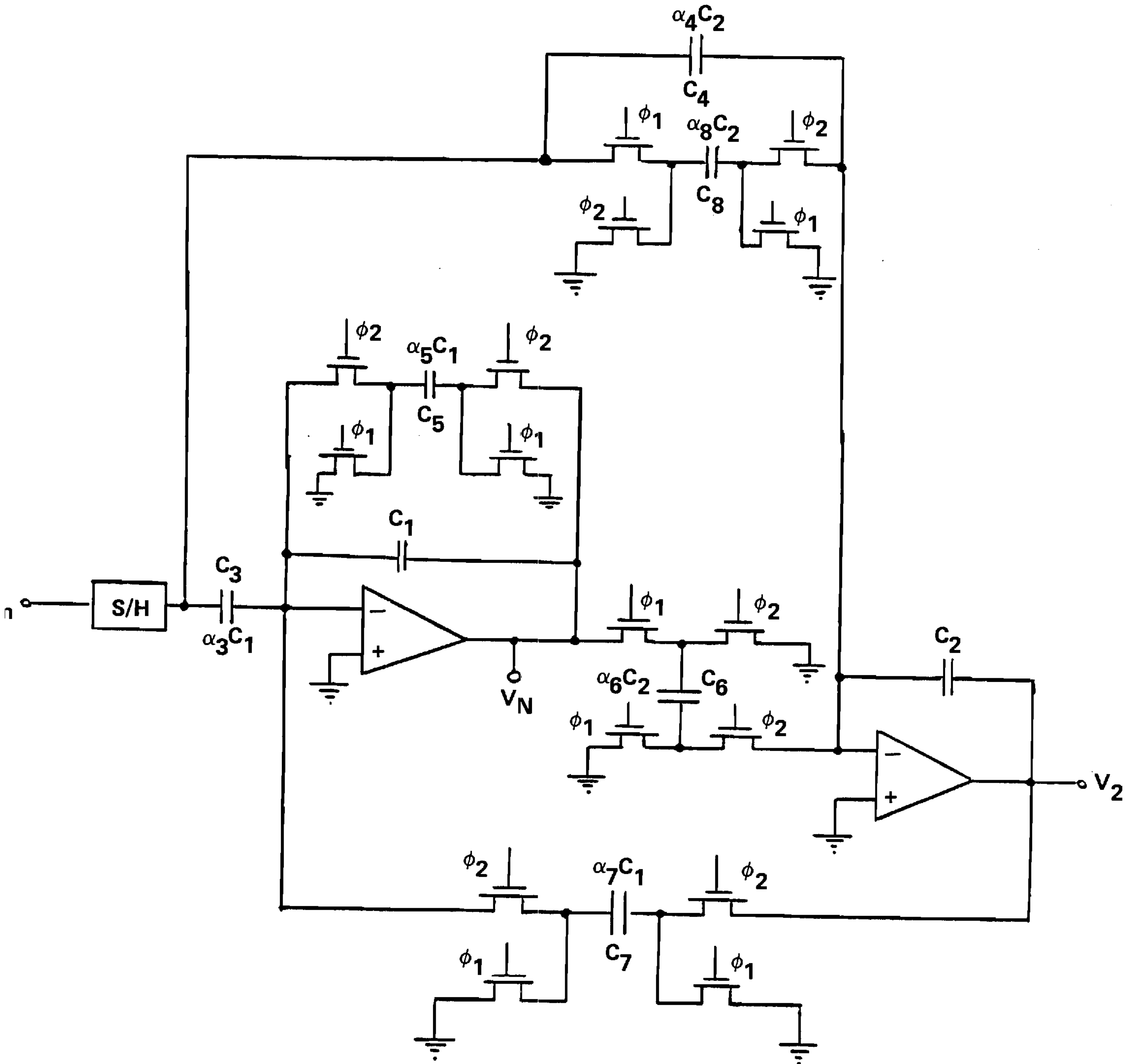


Fig. 5.2 Switched–capacitor implementation of a second–order notch filter.

$$\alpha_4 = \frac{\alpha_3 B_n}{\omega_d \sqrt{\rho}} = \frac{B_n}{\omega_d \sqrt{\rho}} \tag{5-7}$$

$$\alpha_5 = B_d T_C \tag{5-8}$$

$$\alpha_6 = \frac{\omega_d \ T_C}{\sqrt{\rho}} \tag{5-9}$$

$$\alpha_7 = \rho \alpha_6 \tag{5-10}$$

$$\alpha_8 = \frac{\alpha_3 \ \omega_n^2 T_C}{\omega_d \sqrt{\rho}} \tag{5-11}$$

All eight capacitors in Fig. 5.2 can be determined when the ratio between any two is specified and when the value of the minimum acceptable capacitance, C_{min}, is set. Therefore define

$$\alpha_C = C_2/C_1 \tag{5-12}$$

Normally the notch depth specification will cause α_4 in Eq. 5-7 to be the smallest of the α values. For this condition

$$C_4 = C_{min} = \alpha_4 C_2 \tag{5-13}$$

(The computer-aided design program compares all α values to determine which is smallest and then sets the appropriate α-capacitor product equal to C_{min}.) Now all remaining capacitance values can be found from the relationships:

$$
\begin{aligned}
C_3 &= \alpha_3 C_1 & \qquad C_2 &= \alpha_C C_1 \\
C_5 &= \alpha_5 C_1 & C_4 &= \alpha_4 C_2 \\
C_7 &= \alpha_7 C_1 & C_6 &= \alpha_6 C_2 \\
& & C_8 &= \alpha_8 C_2
\end{aligned}
\tag{5-14}
$$

Figure 5.3 shows a block diagram representation of the notch filter in terms of the z-operator. The transfer function for the notch output is found to be

$$\frac{V_N}{V_{in}} = -K' \left[\frac{Z^2 + A_1 Z + A_o}{Z^2 + B_1 Z + B_o} \right] \tag{5-15}$$

$$K' = \frac{\alpha_3}{1 + \alpha_5} \tag{5-16}$$

$$A_1 = \alpha_4 \alpha_7 (\alpha_8/\alpha_4 + 1) - 2 \tag{5-17}$$

$$A_o = 1 - \alpha_4 \alpha_7 \tag{5-18}$$

$$B_1 = \frac{\alpha_6 \alpha_7 - \alpha_5 - 2}{1 + \alpha_5} \tag{5-19}$$

$$B_o = \frac{1}{1 + \alpha_5} \tag{5-20}$$

The magnitude and phase of the notch filter can be determined at any arbitrary frequency by making the substitutions for Z, given in Eqs. 3-29 and 3-30, into Eq. 5-15.

B. NOTCH2 Computer-Aided Design Program

NOTCH2 is a computer-aided design program based upon the configuration of Gregorian [5]. The program is called by the following commands:

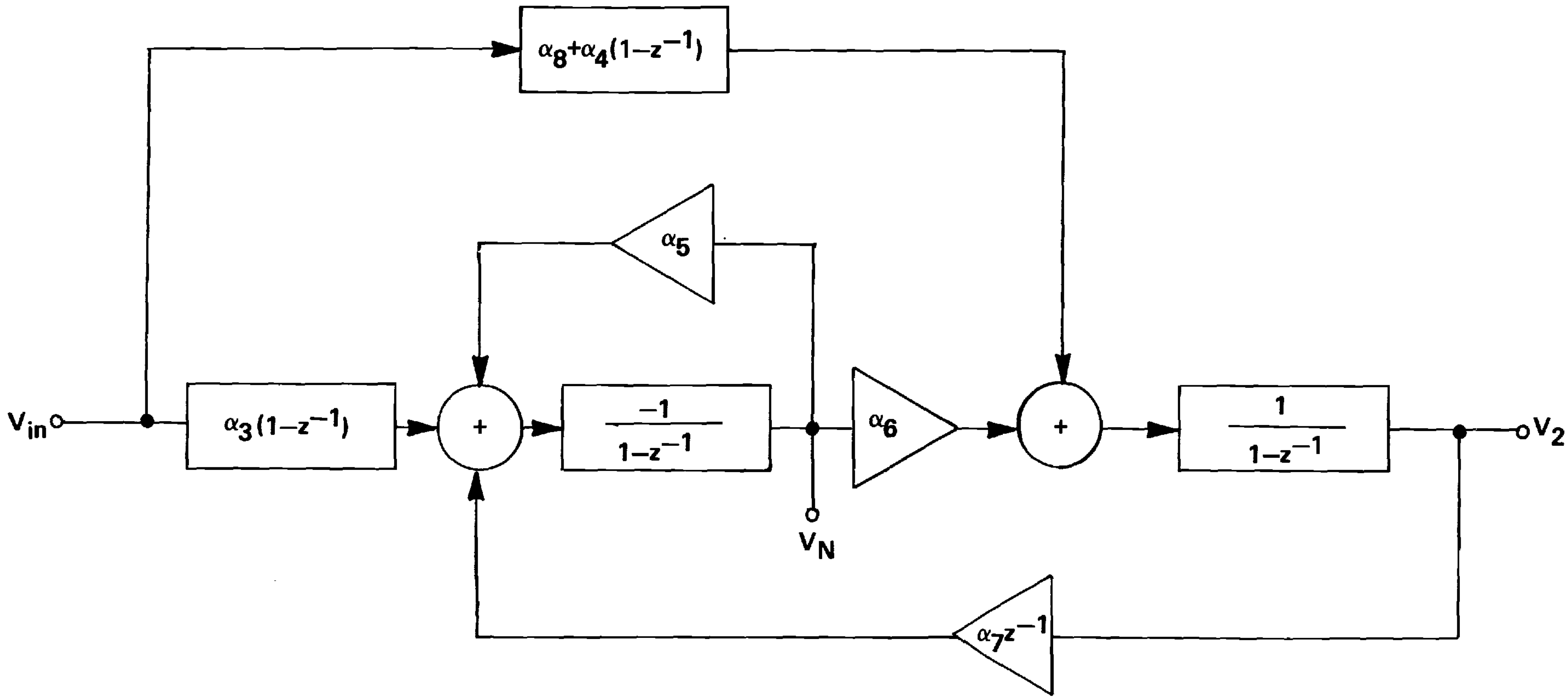


Fig. 5.3 Block diagram representation of the notch filter.

OLD NOTCH2

FORTRAN

RUN-10 NOTCH2

After logging on, the menu of commands will appear as below.

```
SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR
TIMCON NOTCH FILTER. ALGORITHM BY J.A. CONNELLY,
FORTRAN VERSION BY W.N. GOOLSBY.
REVISION 6/9/81.


NEED MENU? (0 =NO, 1 = YES)
=1
A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,
ENTER THE NUMBER OF THE APPROPRIATE COMMAND.


1.  ENTER ALL NEW PARAMETERS
2.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F
3.  FIND CAPACITOR VALUES AND TOTAL C
4.  PRINT MENU
5.  TERMINATE PROGRAM
6.  WARP FN AND BWD.
7.  PRINT CURRENT PARAMETER VALUES.
8.  FIND MAGNITUDE AND PHASE AT A SERIES
    OF FREQUENCY POINTS.
9.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON FN & BWD
10.  CHANGE FN ONLY
11.  CHANGE FD ONLY
12.  CHANGE BWN ONLY
13.  CHANGE BWD ONLY
14.  CHANGE RHO ONLY
15.  CHANGE FC ONLY
16.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER
```

Utilization of the various menu commands follows the same procedure as previously discussed for MARTLP and MARTBP. A design example is given as a further illustration. A complete listing of the NOTCH2 program is given in Appendix C.

C. NOTCH2 Design Example.

```
ENTER COMMAND NUMBER (#4 FOR MENU)
=1
ENTER FN (HERTZ)
=3000
ENTER FD (HERTZ)
=3000
ENTER BWN (HERTZ)
=6.4
ENTER BWD (HERTZ)
=200
ENTER RHO
=.07
ENTER FC (HERTZ)
=200000

ENTER COMMAND NUMBER (#4 FOR MENU)
=3
ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF
=1
ENTER RATIO C2/C1
=.8
ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL
=.2
FOR MINIMUM C=   1.0000000      PF
AND C2/C1=      0.80000000 ,THE CAPACITANCE VALUES ARE:

C(1)=   159.15494      PF
C(2)=   127.32395      PF
C(3)=   159.15494      PF
C(4)=   1.0266439      PF
C(5)=   1.0000000      PF
C(6)=   45.355737      PF
C(7)=   3.9686270      PF
C(8)=   45.355737      PF

TOTAL CAPACITANCE USED=   542.34059      PF
AT  0.2000     PF/SQ MIL, AREA USED =   2711.70      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=16
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  RHO
=2

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.07,.08,.0001
PARAMETER NAME:
RHO
FOR PARAMETER=      0.07370000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   540.11168      PF
```

```
THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=16
TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE
PARAMETER FROM THE FOLLOWING LIST:
     1.  ALPHAC=C2/C1
     2.  RHO
=1

ENTER IN ORDER, SEPARATED BY COMMAS:
PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT
=.7,.9,.01
PARAMETER NAME:
ALPHAC
FOR PARAMETER=       0.80000000  ,
THE SMALLEST VALUE OF TOTAL CAP. =   540.11168      PF


THE OPTIMIZED PARAMETER HAS BEEN PLACED
IN THE ARGUMENT LIST.

ENTER COMMAND NUMBER (#4 FOR MENU)
=7
VALUES AS CURRENTLY CALCULATED ARE:
FN=   3000.0000      HERTZ
FD=   3000.0000      HERTZ
BWN=   6.4000000      HERTZ
BWD=   200.00000      HERTZ
RHO=      0.07370000  (DIMENSIONLESS)
ALPHAC=      0.80000000  (DIMENSIONLESS)
FC=   200000.00      HERTZ
T=  0.50000000E-05  SECONDS
ALPHA(3)=   1.0000000
ALPHA(4)=  0.78582343E-02
ALPHA(5)=  0.62831853E-02
ALPHA(6)=  0.34716616
ALPHA(7)=  0.25586146E-01
ALPHA(8)=  0.34716616
C(1)=   159.15494      PF
C(2)=   127.32395      PF
C(3)=   159.15494      PF
C(4)=   1.0005415      PF
C(5)=   1.0000000      PF
C(6)=   44.202568      PF
C(7)=   4.0721616      PF
C(8)=   44.202568      PF
MINIMUM C =   1.0000000      PF
TOTAL C =   540.11168      PF
AT  0.2000     PF/SQ MIL, AREA USED =   2700.56      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=6
FINDS PREWARPED VALUES FOR FN AND BWD.
YOU MAY CHOOSE TO WARP FN TO MAGNITUDE PEAK
OR TO 180 DEGREE PHASE CROSSING.
ENTER NUMBER OF DESIRED COMMAND.
     1.  MAGNITUDE PEAK
     2.  180 DEGREE PHASE CROSSING
=2
```

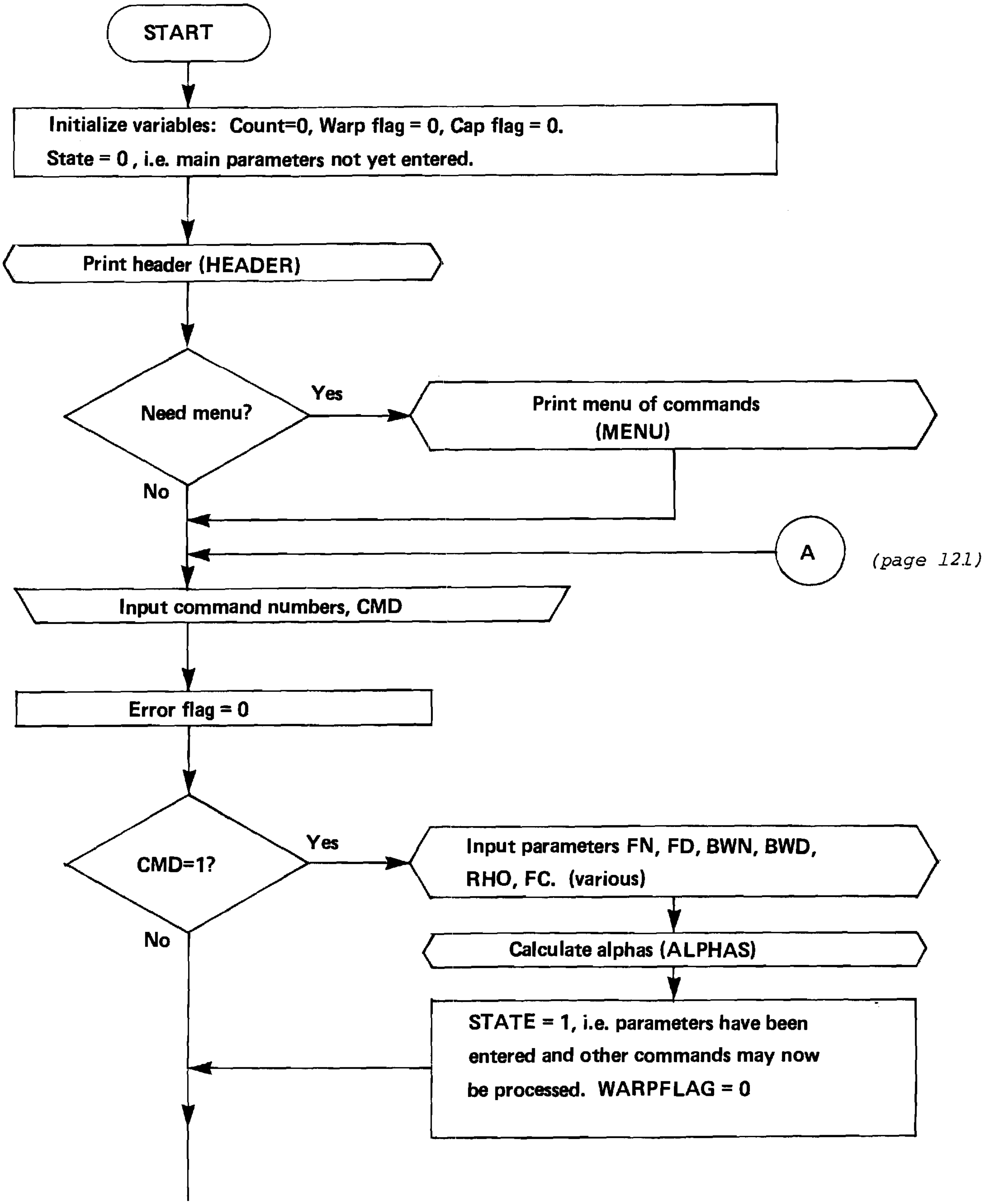
```
DESIRED NOTCH CENTER=   3000.0000    HZ
DESIRED BANDWIDTH=   200.00000    HZ
PREWARPED VARIABLES ARE:  FN=   2998.6257    HZ
BWD=   199.30505    HZ
-3DB POINTS WERE FOUND AT:
F1=   2905.0374        AND F2=   3105.0480      HERTZ
WARP ITERATION COUNT=   3 CYCLES
FINDPK ITERATION COUNT=   166  CYCLES.

ENTER COMMAND NUMBER (#4 FOR MENU)
=7
VALUES AS CURRENTLY CALCULATED ARE:
FN=   2998.6257      HERTZ
FD=   3000.0000      HERTZ
BWN=   6.4000000      HERTZ
BWD=   199.30505      HERTZ
RHO=       0.07370000  (DIMENSIONLESS)
ALPHAC=       0.80000000  (DIMENSIONLESS)
FC=   200000.00      HERTZ
T=  0.50000000E-05  SECONDS
-3DB POINTS ARE LOCATED AT FREQUENCIES:
F1=   2905.0374        AND F2=   3105.0480      HERTZ
ALPHA(3)=   1.0000000
ALPHA(4)=  0.78582343E-02
ALPHA(5)=  0.62613527E-02
ALPHA(6)=  0.34716616
ALPHA(7)=  0.25586146E-01
ALPHA(8)=  0.34684817
C(1)=   159.70990      PF
C(2)=   127.76792      PF
C(3)=   159.70990      PF
C(4)=   1.0040302      PF
C(5)=   1.0000000      PF
C(6)=   44.356697      PF
C(7)=   4.0863607      PF
C(8)=   44.316068      PF
MINIMUM C =   1.0000000      PF
TOTAL C =   541.95087      PF
AT  0.2000     PF/SQ MIL, AREA USED =   2709.75      SQ MILS

ENTER COMMAND NUMBER (#4 FOR MENU)
=8
CALCULATES MAGNITUDE AND PHASE AT A SERIES
OF FREQUENCY POINTS.  ENTER STARTING AND ENDING
FREQUENCIES (HERTZ), SEPARATED BY A COMMA.
=2995,3005
ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)
=1
```

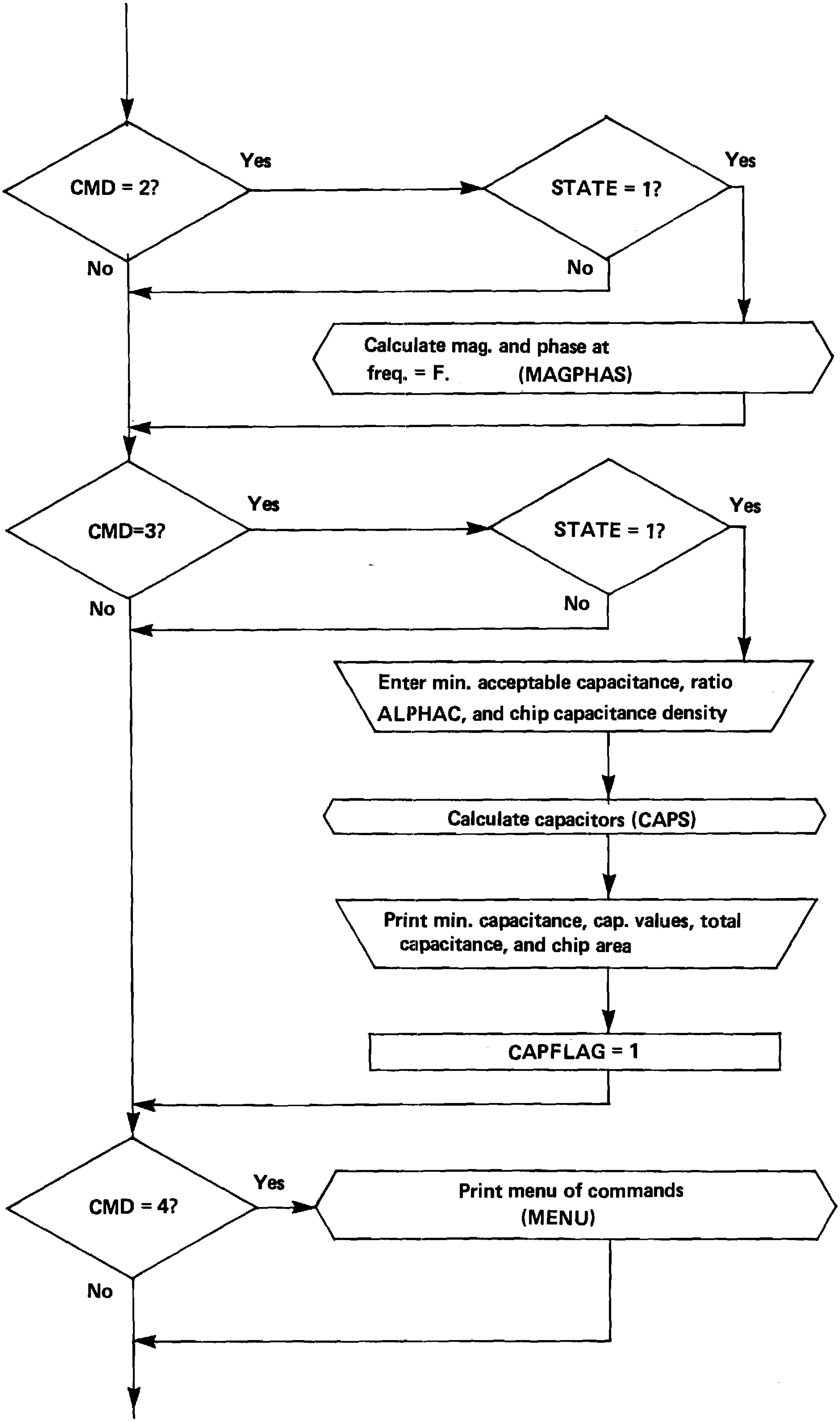
FREQUENCY (HZ)	MAGNITUDE (DB)	MAGNITUDE	PHASE (DEG)
2995.0000	-24.636430	0.58637913E-01	124.02022
2996.0000	-25.928037	0.50535685E-01	129.69396
2997.0000	-27.279796	0.43252397E-01	137.60339
2998.0000	-28.571876	0.37274015E-01	148.56295
2999.0000	-29.548486	0.33310082E-01	163.02759
3000.0000	-29.866939	0.32110942E-01	179.96811
3001.0000	-29.379911	0.33962875E-01	196.54333
3002.0000	-28.309473	0.38417257E-01	210.21965
3003.0000	-26.995497	0.44691522E-01	220.41214
3004.0000	-25.659782	0.52120779E-01	227.74038
[illegible]	[illegible]	[illegible]	[illegible]

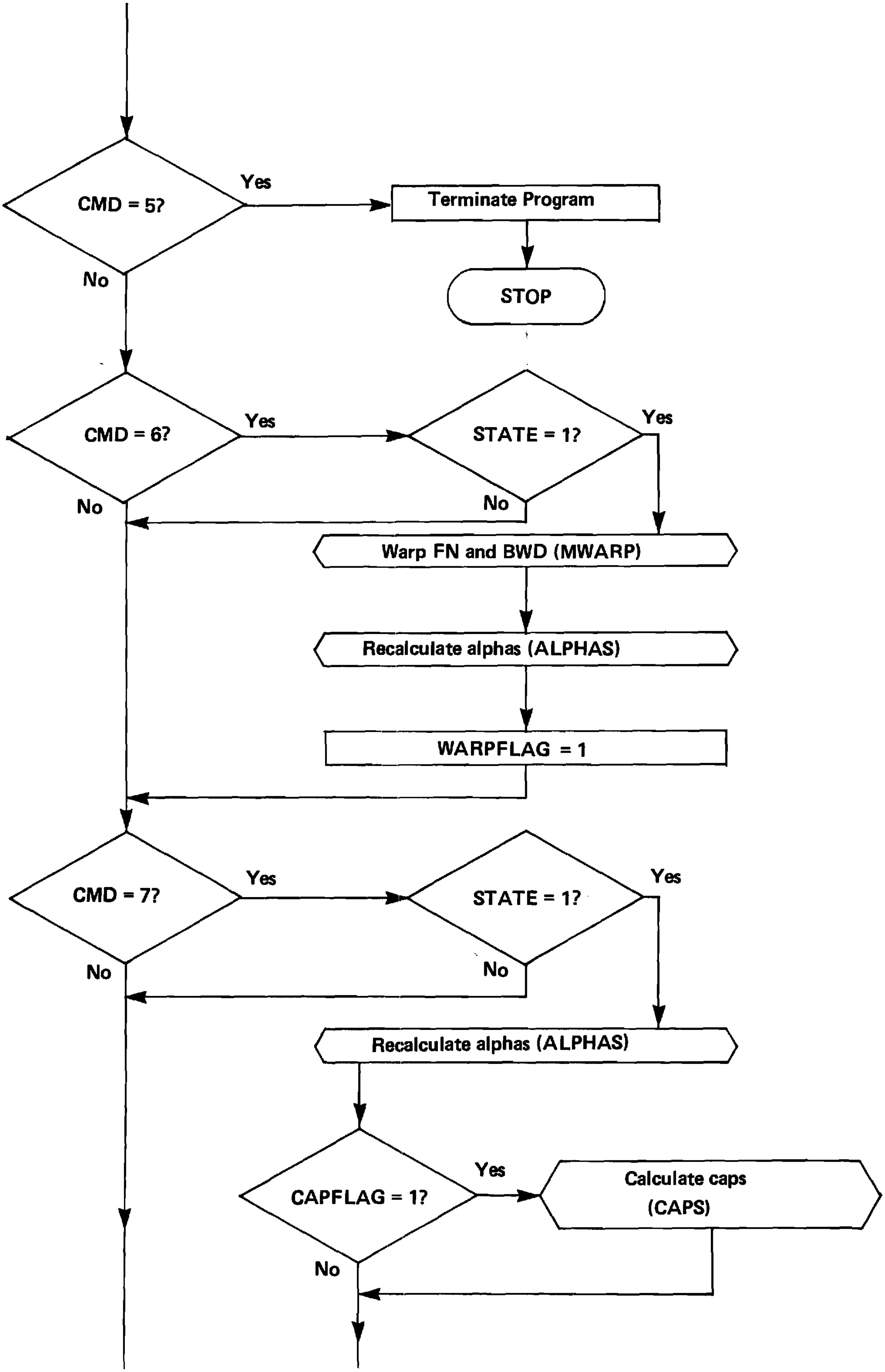
D. Flow Chart for NOTCH2.

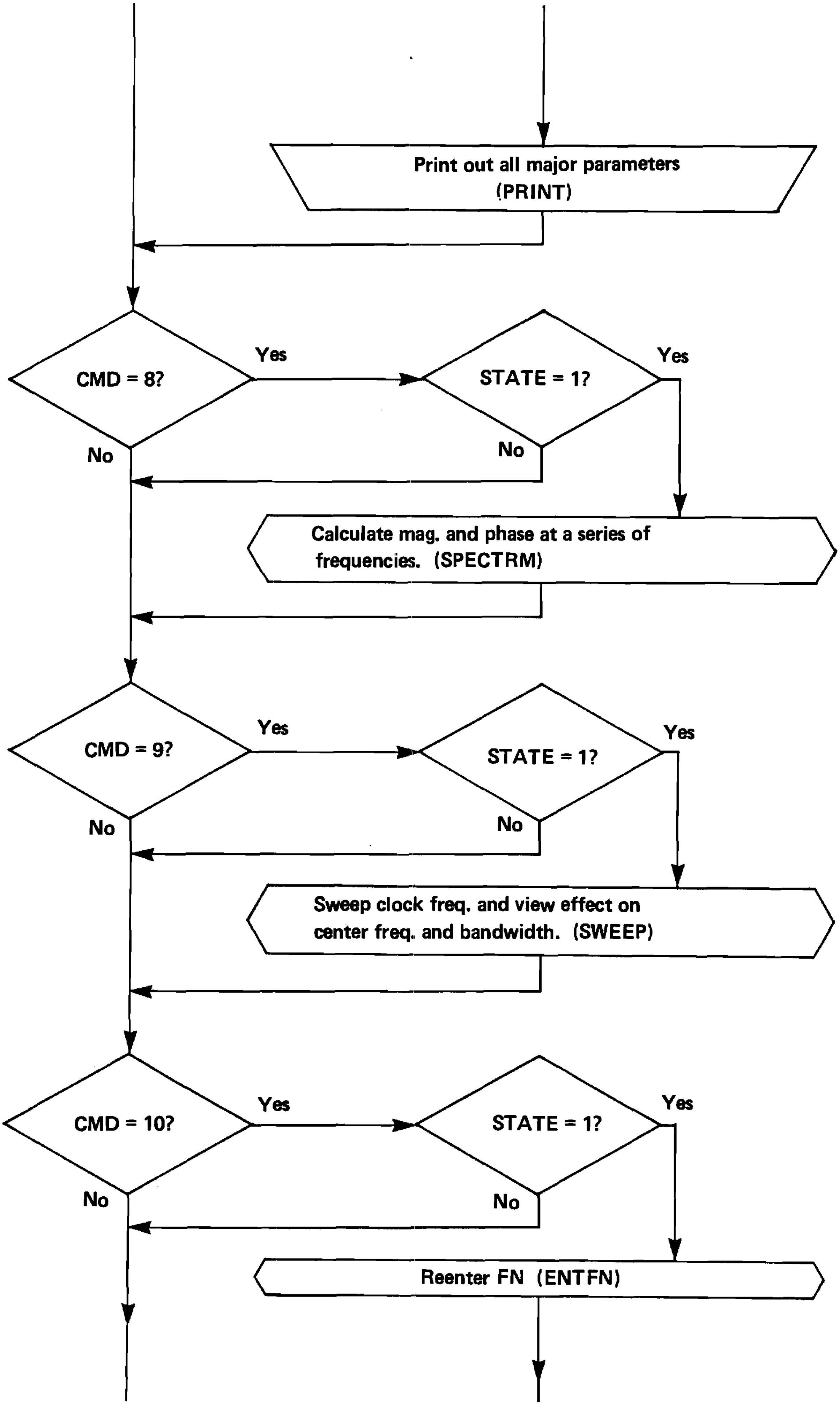
Program NOTCH2

Main Program

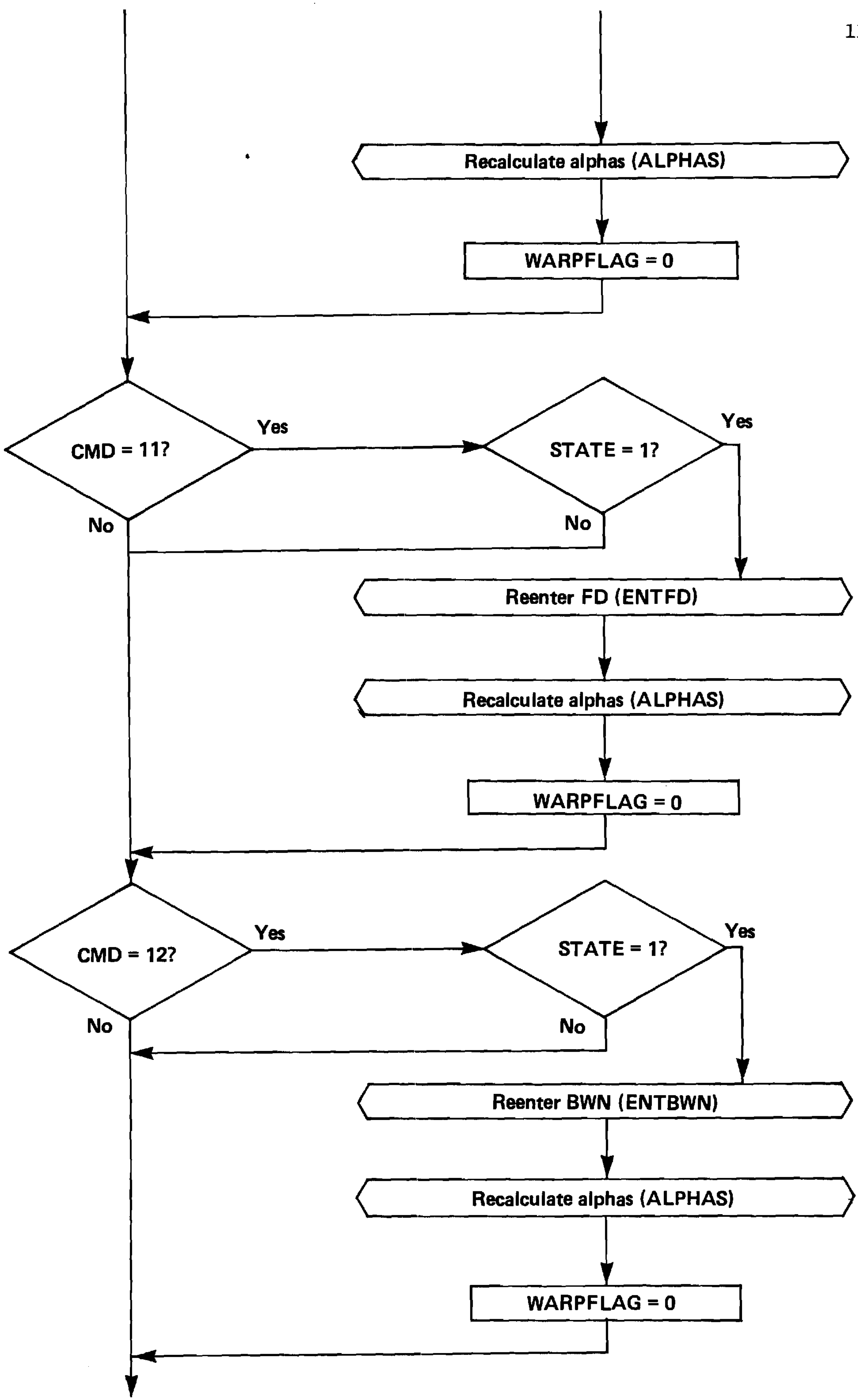


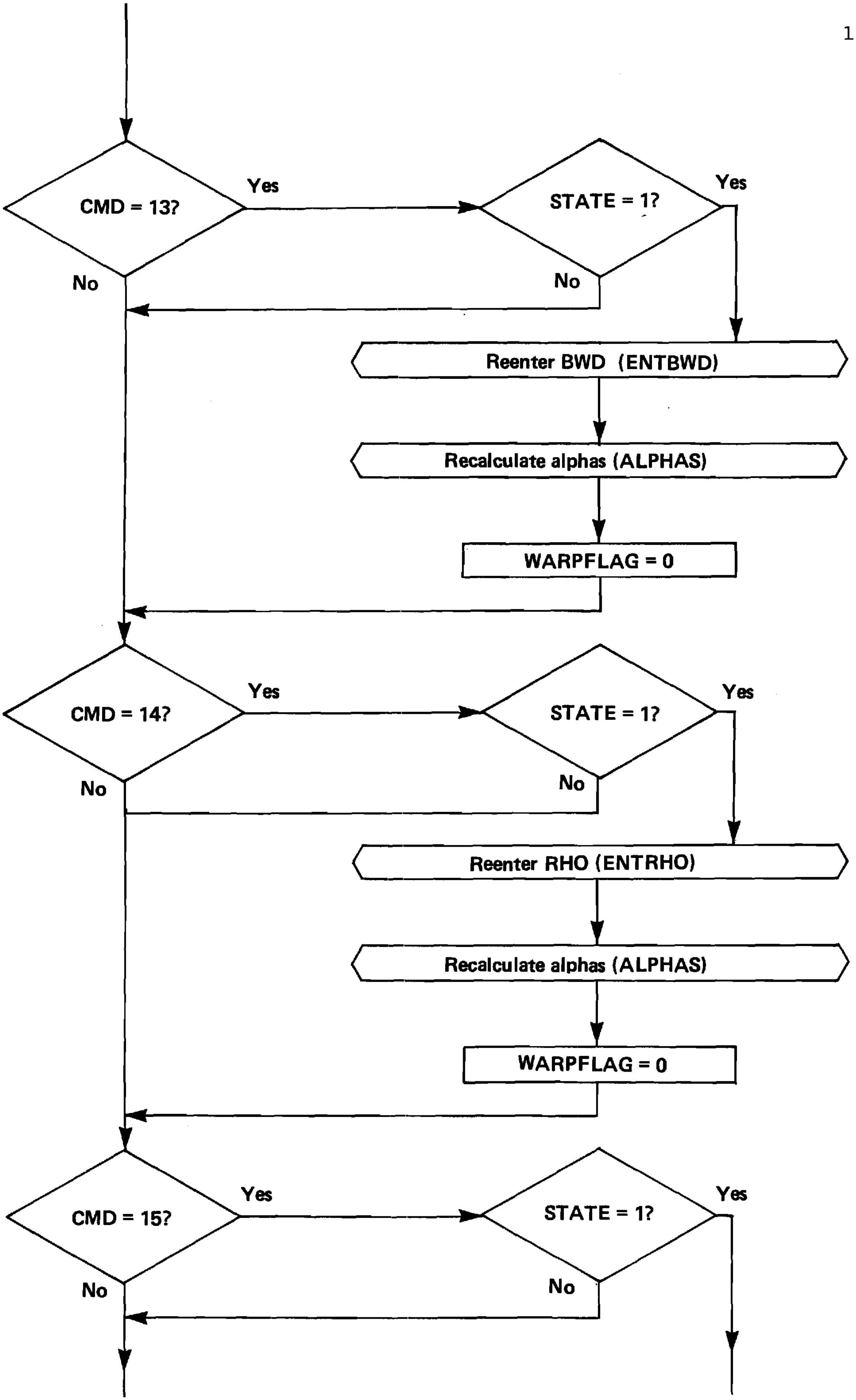

CMD = 2?
Yes
No
STATE = 1?
Yes
No
Calculate mag. and phase at freq. = F. (MAGPHAS)
CMD=3?
Yes
No
STATE = 1?
Yes
No
Enter min. acceptable capacitance, ratio ALPHAC, and chip capacitance density
Calculate capacitors (CAPS)
Print min. capacitance, cap. values, total capacitance, and chip area
CAPFLAG = 1
CMD = 4?
Yes
No
Print menu of commands (MENU)

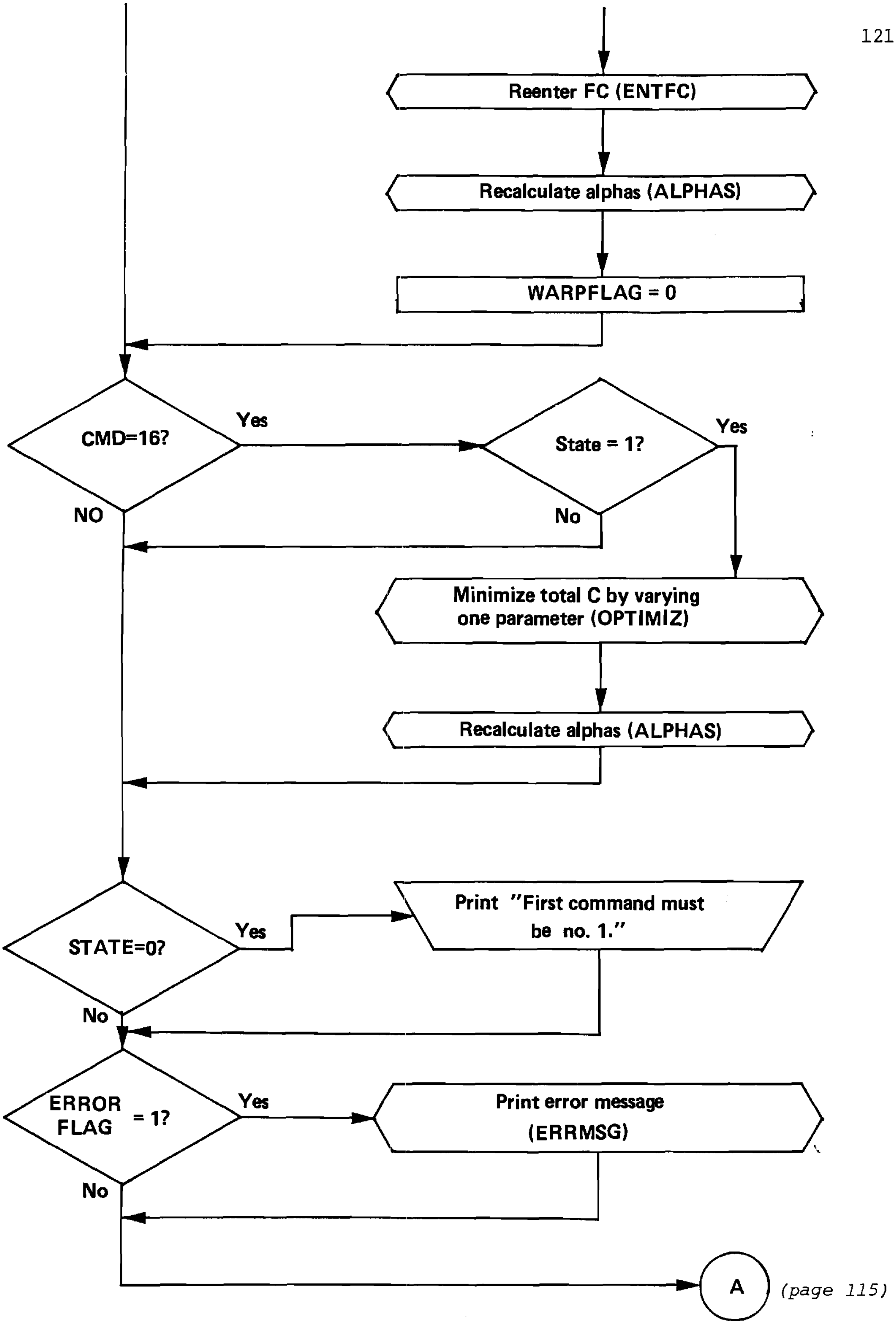

CMD = 5?
Yes
No
Terminate Program
STOP
CMD = 6?
Yes
No
STATE = 1?
Yes
No
Warp FN and BWD (MWARP)
Recalculate alphas (ALPHAS)
WARPFLAG = 1
CMD = 7?
Yes
No
STATE = 1?
Yes
No
Recalculate alphas (ALPHAS)
CAPFLAG = 1?
Yes
No
Calculate caps
(CAPS)



Print out all major parameters
(PRINT)
CMD = 8?
Yes
STATE = 1?
Yes
No
No
Calculate mag. and phase at a series of
frequencies. (SPECTRM)
CMD = 9?
Yes
STATE = 1?
Yes
No
No
Sweep clock freq. and view effect on
center freq. and bandwidth. (SWEEP)
CMD = 10?
Yes
STATE = 1?
Yes
No
No
Reenter FN (ENTFN)

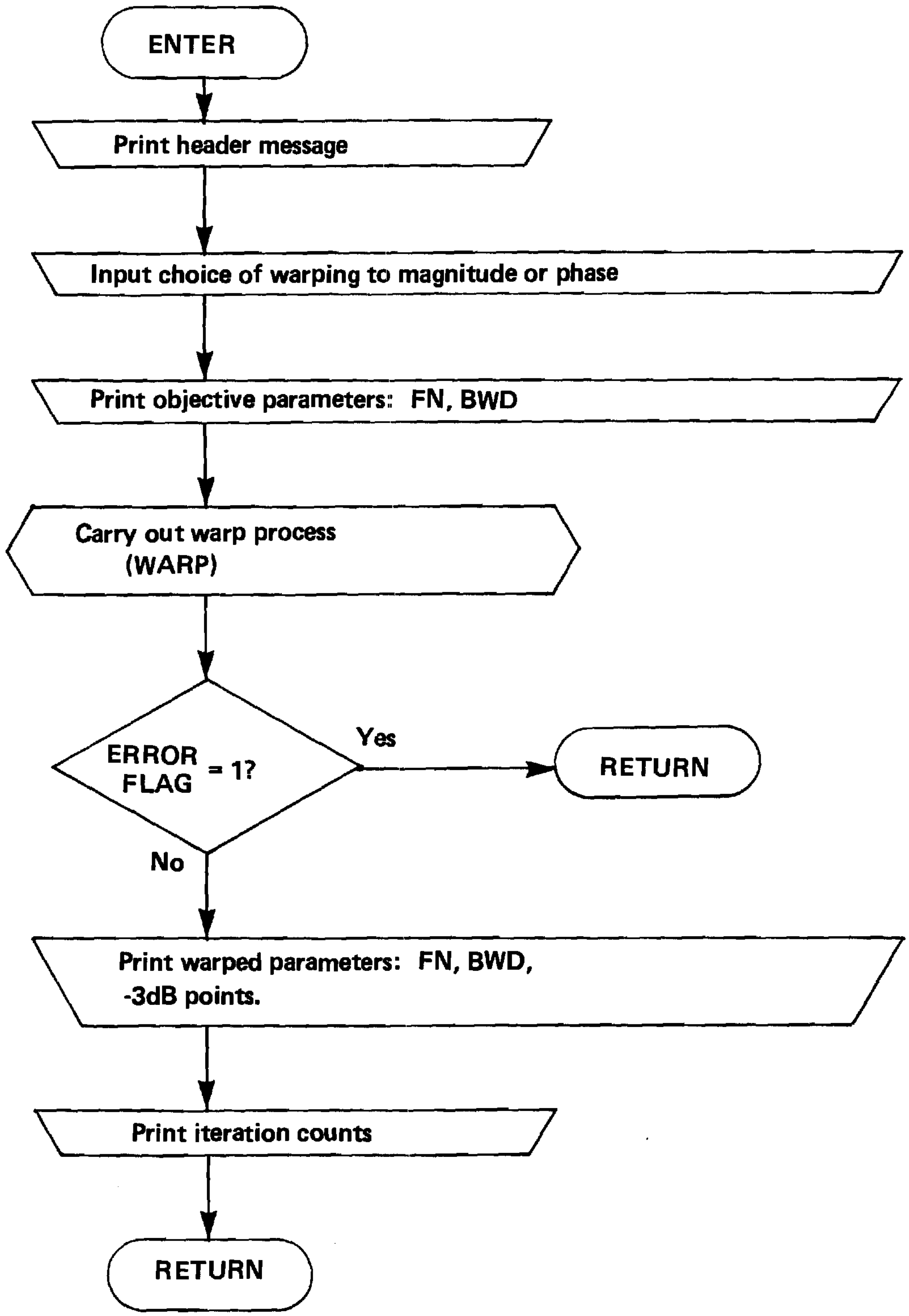

Recalculate alphas (ALPHAS)
WARPFLAG = 0
CMD = 11?
Yes
No
STATE = 1?
Yes
No
Reenter FD (ENTFD)
Recalculate alphas (ALPHAS)
WARPFLAG = 0
CMD = 12?
Yes
No
STATE = 1?
Yes
No
Reenter BWN (ENTBWN)
Recalculate alphas (ALPHAS)
WARPFLAG = 0

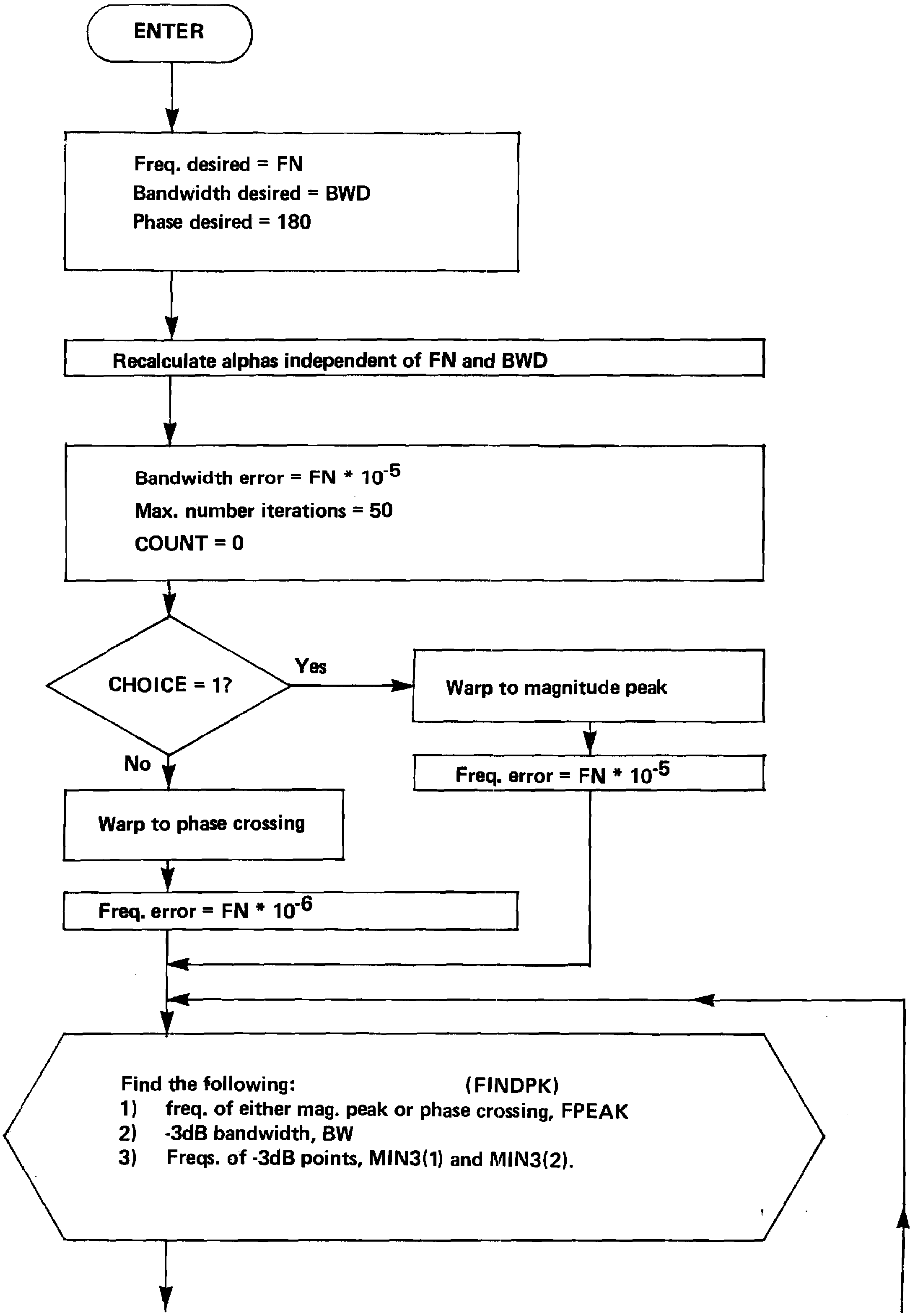

CMD = 13?
Yes
No
STATE = 1?
Yes
No
Reenter BWD (ENTBWD)
Recalculate alphas (ALPHAS)
WARPFLAG = 0
CMD = 14?
Yes
No
STATE = 1?
Yes
No
Reenter RHO (ENTRHO)
Recalculate alphas (ALPHAS)
WARPFLAG = 0
CMD = 15?
Yes
No
STATE = 1?
Yes
No

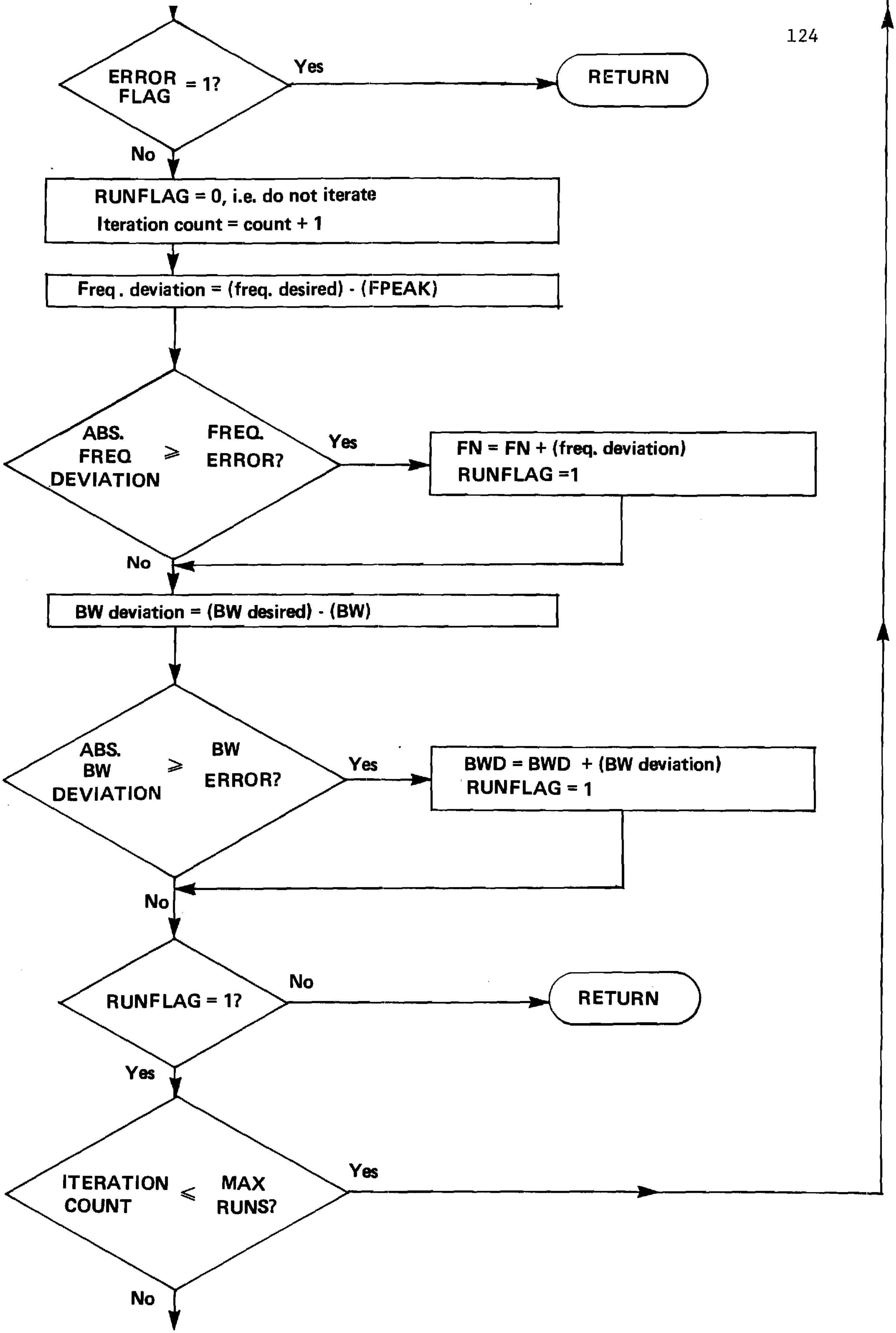

Reenter FC (ENTFC)
Recalculate alphas (ALPHAS)
WARPFLAG = 0
CMD=16?
Yes
NO
State = 1?
Yes
No
Minimize total C by varying one parameter (OPTIMIZ)
Recalculate alphas (ALPHAS)
STATE=0?
Yes
No
Print "First command must be no. 1."
ERROR FLAG = 1?
Yes
No
Print error message (ERRMSG)
A
(page 115)

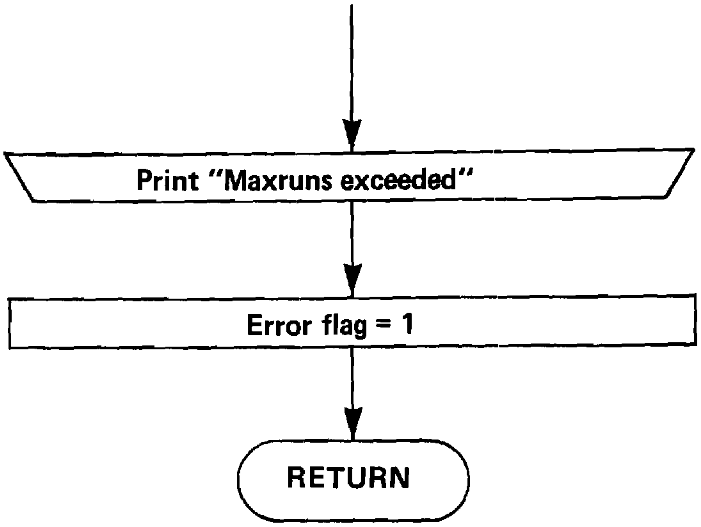
Warping Routines for Program NOTCH2

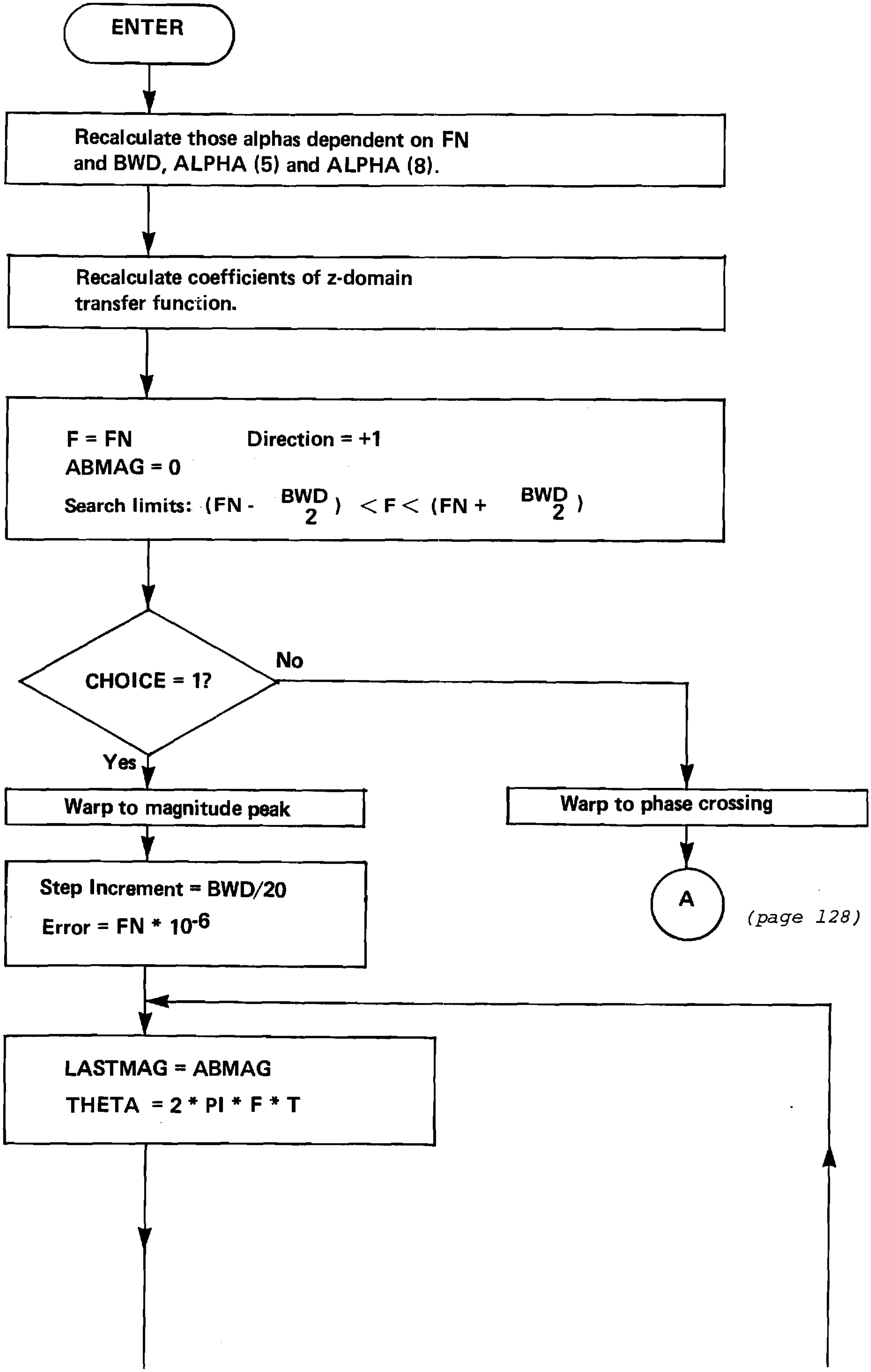
Subroutine MWARP

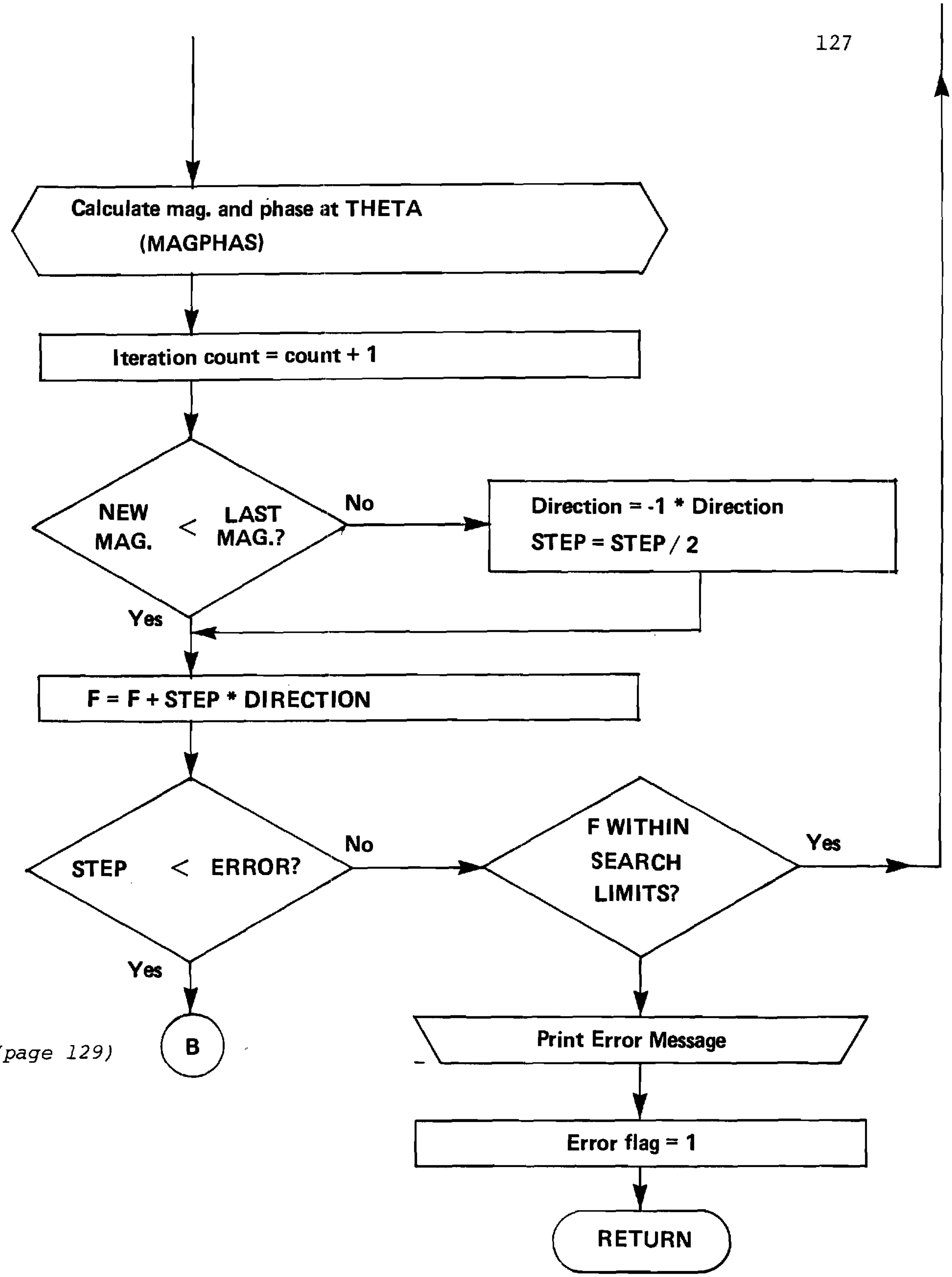



ENTER
Freq. desired = FN
Bandwidth desired = BWD
Phase desired = 180
Recalculate alphas independent of FN and BWD
Bandwidth error = FN * 10^-5
Max. number iterations = 50
COUNT = 0
CHOICE = 1?
Yes
Warp to magnitude peak
Freq. error = FN * 10^-5
No
Warp to phase crossing
Freq. error = FN * 10^-6
Find the following:
(FINDPK)
1) freq. of either mag. peak or phase crossing, FPEAK
2) -3dB bandwidth, BW
3) Freqs. of -3dB points, MIN3(1) and MIN3(2).

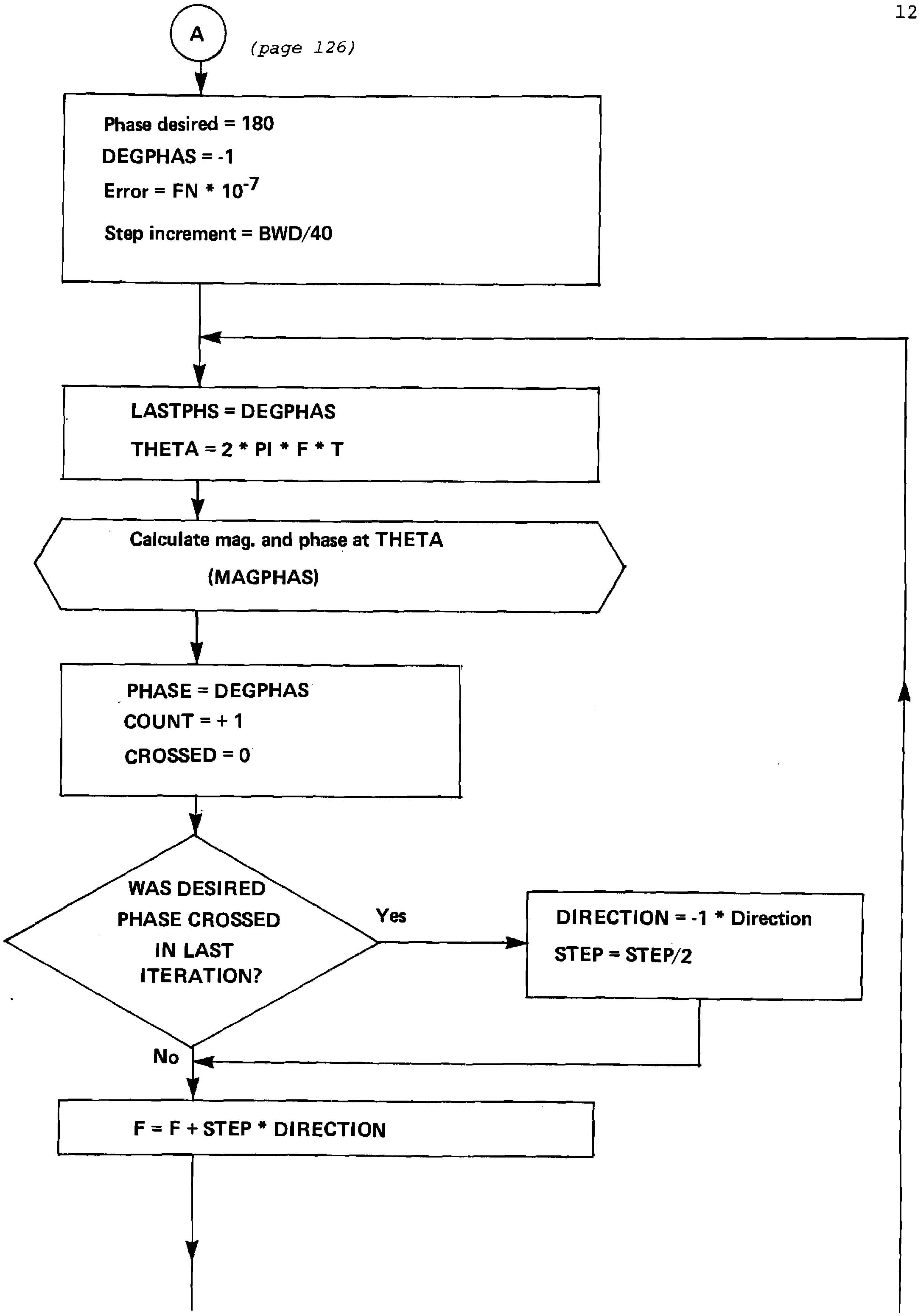

ERROR FLAG = 1?
Yes
RETURN
No
RUNFLAG = 0, i.e. do not iterate
Iteration count = count + 1
Freq . deviation = (freq. desired) - (FPEAK)
ABS. FREQ DEVIATION ⩾ FREQ. ERROR?
Yes
FN = FN + (freq. deviation)
RUNFLAG =1
No
BW deviation = (BW desired) - (BW)
ABS. BW DEVIATION ⩾ BW ERROR?
Yes
BWD = BWD + (BW deviation)
RUNFLAG = 1
No
RUNFLAG = 1?
No
RETURN
Yes
ITERATION COUNT ⩽ MAX RUNS?
Yes
No

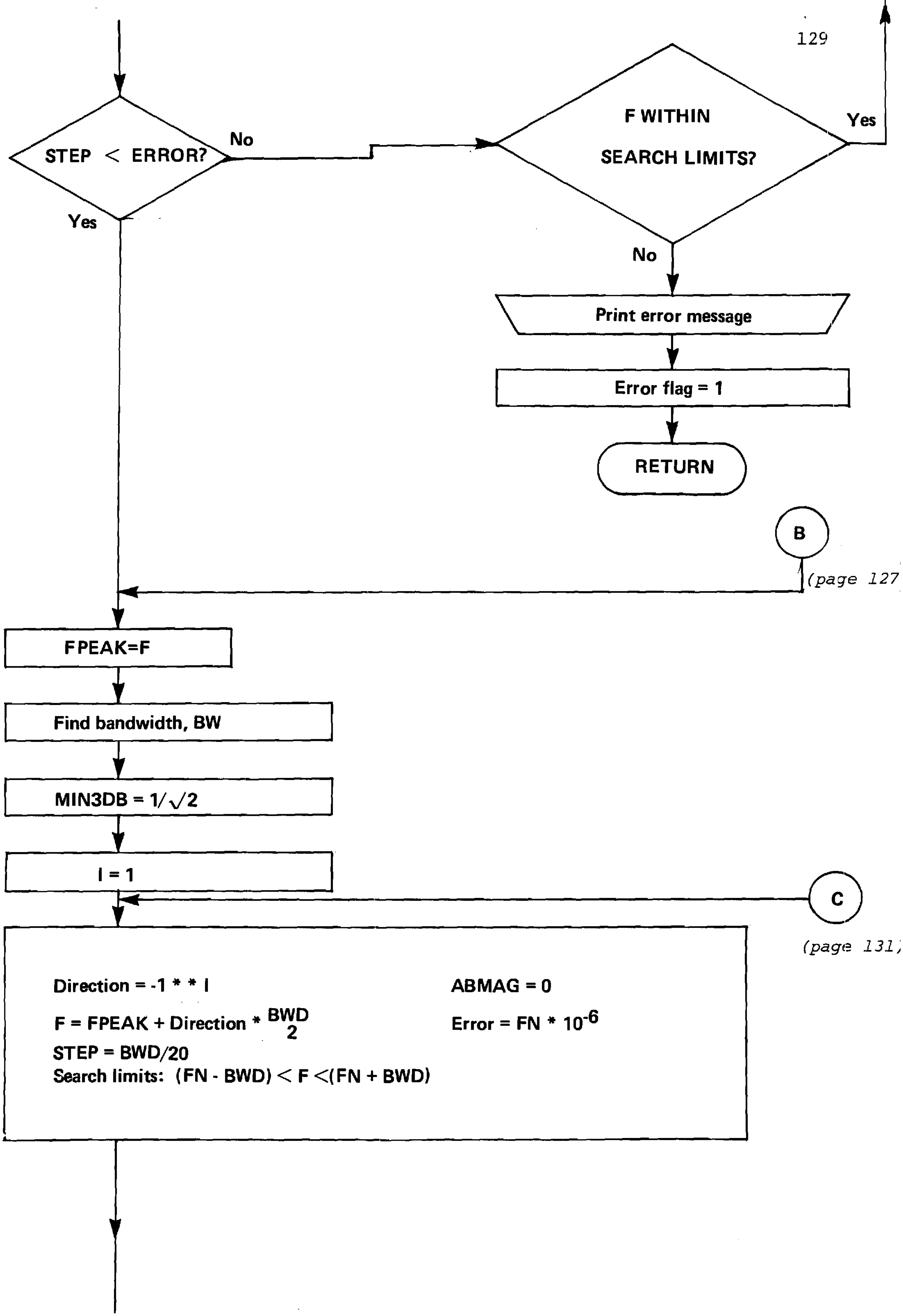

Print "Maxruns exceeded"
Error flag = 1
RETURN

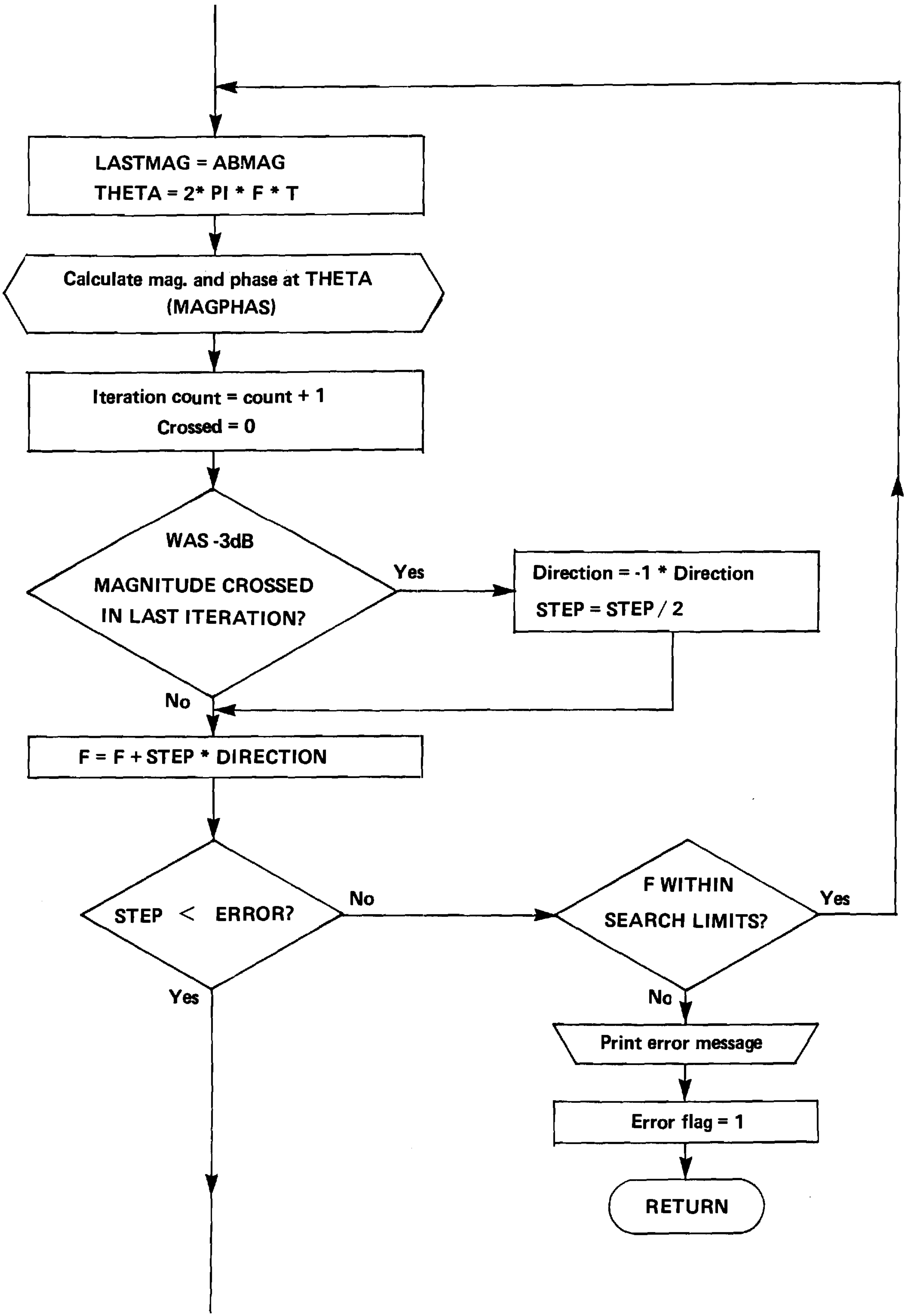

ENTER
Recalculate those alphas dependent on FN and BWD, ALPHA (5) and ALPHA (8).
Recalculate coefficients of z-domain transfer function.
F = FN
Direction = +1
ABMAG = 0
Search limits: (FN - BWD/2) < F < (FN + BWD/2)
CHOICE = 1?
No
Yes
Warp to magnitude peak
Warp to phase crossing
A
(page 128)
Step Increment = BWD/20
Error = FN * 10-6
LASTMAG = ABMAG
THETA = 2 * PI * F * T

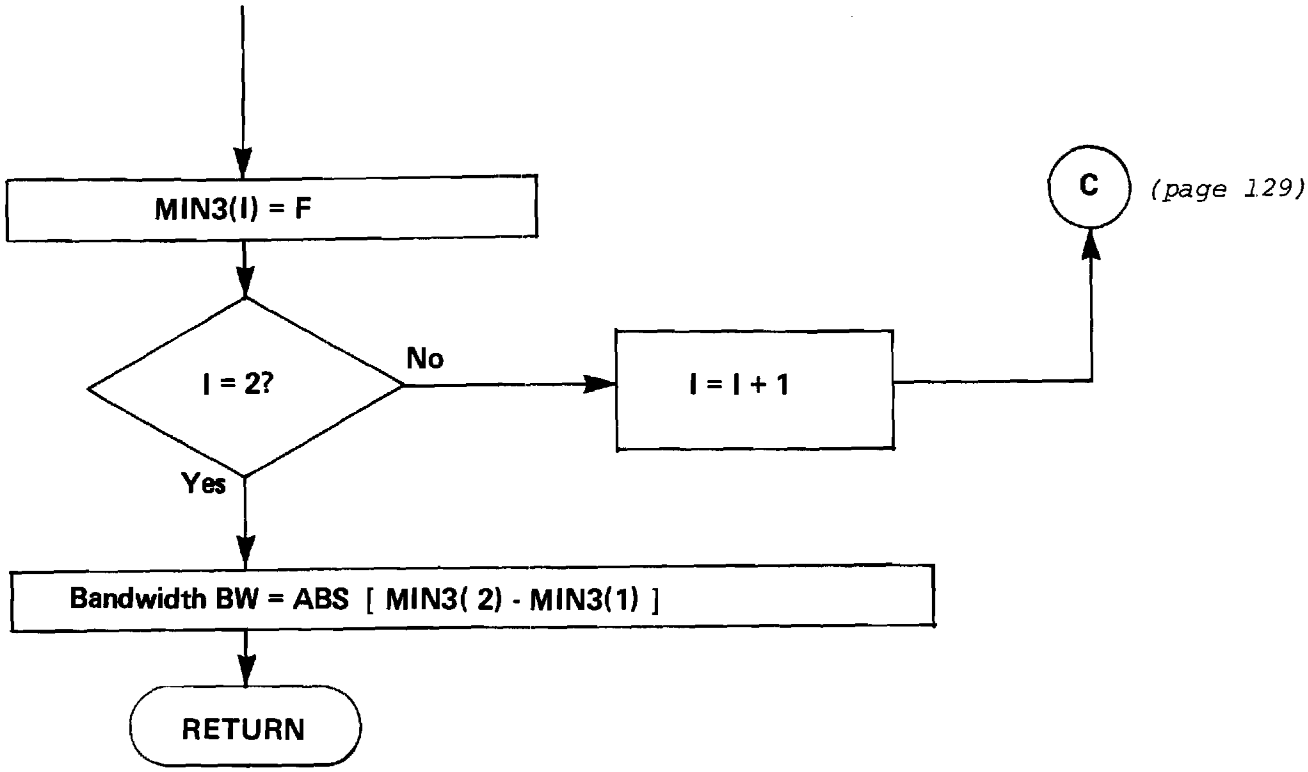

Calculate mag. and phase at THETA (MAGPHAS)
Iteration count = count + 1
NEW MAG. < LAST MAG.?
No
Yes
Direction = -1 * Direction
STEP = STEP / 2
F = F + STEP * DIRECTION
STEP < ERROR?
No
Yes
F WITHIN SEARCH LIMITS?
Yes
B
Print Error Message
Error flag = 1
RETURN

(page 129)


A
(page 126)
Phase desired = 180
DEGPHAS = -1
Error = FN * 10^-7
Step increment = BWD/40
LASTPHS = DEGPHAS
THETA = 2 * PI * F * T
Calculate mag. and phase at THETA
(MAGPHAS)
PHASE = DEGPHAS
COUNT = + 1
CROSSED = 0
WAS DESIRED PHASE CROSSED IN LAST ITERATION?
Yes
DIRECTION = -1 * Direction
STEP = STEP/2
No
F = F + STEP * DIRECTION


STEP < ERROR?
No
Yes
F WITHIN
SEARCH LIMITS?
Yes
No
Print error message
Error flag = 1
RETURN
B
(page 127)
FPEAK=F
Find bandwidth, BW
MIN3DB = 1/√2
I = 1
C
(page 131)
Direction = -1 * * I
F = FPEAK + Direction * BWD/2
STEP = BWD/20
Search limits: (FN - BWD) < F <(FN + BWD)
ABMAG = 0
Error = FN * 10^-6


LASTMAG = ABMAG
THETA = 2* PI * F * T
Calculate mag. and phase at THETA
(MAGPHAS)
Iteration count = count + 1
Crossed = 0
WAS -3dB
MAGNITUDE CROSSED
IN LAST ITERATION?
Yes
Direction = -1 * Direction
STEP = STEP / 2
No
F = F + STEP * DIRECTION
STEP < ERROR?
No
F WITHIN
SEARCH LIMITS?
Yes
Yes
No
Print error message
Error flag = 1
RETURN


MIN3(I) = F
I = 2?
No
Yes
I = I + 1
C
(page 129)
Bandwidth BW = ABS [MIN3(2) - MIN3(1)]
RETURN

VI. LADDER NETWORKS

A. Biquad Structures

Switched-capacitor filters are sampled-data systems which must be analyzed in the z-domain. However, SCFs are generally synthesized in the s-domain since knowledge of s-domain filters is quite mature and since relatively easy design methods have been developed for the s-domain.

One of the most commonly used techniques in synthesizing arbitrary frequency-selective filters is to cascade second order biquads. Each biquad implements one complex conjugate pole pair and either none, one, or a conjugate pair of zeros. As many biquads as needed are cascaded to produce a filter of the desired order. This design method possesses several well known advantages. First, closed form expressions exist for Butterworth and Chebyshev filters which give ω_o and Q for each pole pair in the desired filter. That is, higher order maximally flat and equiripple filters are easily factored into biquad sections. Second, high quality biquad circuits exist which exhibit low sensitivities and allow easy tuning; premier among these are the three-amplifier state-variable circuits. Thus the cascaded biquad approach is straightforward to implement and produces good to excellent results, at least in filters of low to medium order.

B. Ladder Structures

Another technique for synthesizing s-domain filters is the ladder approach, in which a classical LC passive ladder is first generated, then converted to an active RC circuit by one of several methods. The ladder approach can offer lower component sensitivities than cascaded biquad structures and is thus attractive in situations demanding high precision and/or when designing filters of high order or high Q. Ladder design typically consists of the following steps:

1. Determination of the s-domain transfer function;
2. Determination of the ladder topology and element values for a low-pass prototype (LPP) ladder;
3. Transformation of LPP to the desired filter type, i.e. high-pass, bandpass, or notch;
4. Conversion of the transformed passive ladder to an active RC circuit; and
5. Conversion of the active RC circuit to a switched-capacitor circuit.

Of course, if a low-pass filter is desired, step 3 should be deleted. This scheme represents only one possible design method. In particular, one may wish to replace steps 2 and 3 with a direct ladder design of the desired filter type, because low-pass to bandpass and low-pass to notch frequency transformations are only capable of producing even-order filters with pass-bands geometrically symmetric about the center frequency.

C. Example of Ladder Network Synthesis

To illustrate the design method above, a second-order notch filter will be generated. The desired filter parameters are:

Filter type: Notch   Order (N): 2

$\omega_o = 2\pi \times 3000$ Hz $B = 2\pi \times 200$ Hz

Passband: maximally flat

Finite notch depth; rejection ratio of 30 dB

Step one is to generate the s-domain transfer function. For a second-order notch with finite rejection, we have

$$\frac{V_o(s)}{V_{in}(s)} = K\left[\frac{s^2 + \frac{\omega_o}{Q_n}\ s + \omega_o^2}{s^2 + \frac{\omega_o}{Q_d}\ s + \omega_o^2}\right] \tag{6-1}$$

where

K = asymptotic gain at DC and infinity

ω_o = center frequency in radians

$\frac{\omega_o}{Q_d}$ = -3 dB bandwidth in radians

$\frac{Q_n}{Q_d}$ = rejection ratio

Incidentally, for infinite rejection $Q_n \to \infty$. Step 2 finds a LPP ladder which can be frequency transformed to achieve the notch function. Frequency-band transformations are summarized in Fig. 6.1, where it is seen that the LPP-to-band reject transformation is

$$s = \frac{pB}{p^2 + \omega_o^2} \tag{6-2}$$

where s is the complex frequency variable in the LPP domain, p the complex frequency variable in the band-reject (BR) domain, and $B = \omega_o/Q_d$ is the -3 dB radian bandwidth. Since the LPP-to-BR transformation doubles the order of the filter, we start with a first order LPP. Also, since the notch specifications call for maximally flat pass-band response, the LPP should be of Butterworth type.

Fig. 6.2 shows the canonical realizations of low-pass ladders. For minimum inductance Butterworth ladders the element values are given by

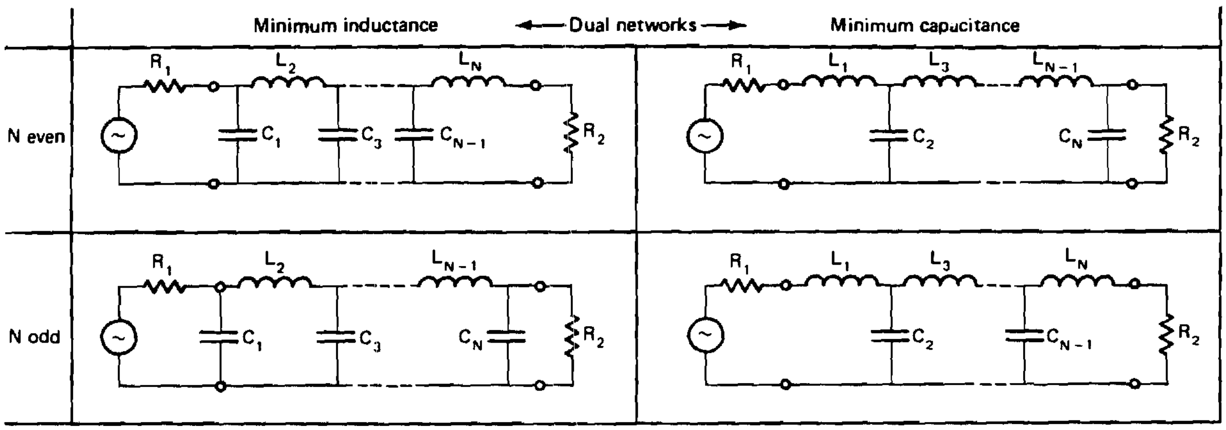


Fig. 6.1 LC ladders for all–pole low–pass filters (Taken from Sedra and Brackett, pg. 206).

Type	Direction of Transformation	Transformation	Notes
LP	LP → LPP	$\Omega_s = \frac{\omega_s}{\omega_p}$	
	LPP → LP	$s = \frac{p}{\omega_p}$	
HP	HP → LPP	$\Omega_s = \frac{\omega_p}{\omega_s}$	
	LPP → HP	$s = \frac{\omega_p}{p}$	
BP	BP → LPP	$\Omega_s = \frac{\omega_{s2} - \omega_{s1}}{\omega_{p2} - \omega_{p1}}$	$\omega_o^2 = \omega_{p1}\,\omega_{p2} = \omega_{s1}\,\omega_{s2}$
	LPP → BP	$s = \frac{p^2 + \omega_o^2}{pB}$	$B = \omega_{p2} - \omega_{p1}$
BR	BR → LPP	$\Omega_s = \frac{\omega_{p2} - \omega_{p1}}{\omega_{s2} - \omega_{s1}}$	Same as above
	LPP → BR	$s = \frac{pB}{p^2 + \omega_o^2}$	

Fig. 6.2 Frequency band transformations (Taken from Sedra and Brackett, pg. 244).

$$\left.\begin{array}{l} k \text{ odd}, \quad C_k \\ \\ k \text{ even}, \quad L_k \end{array}\right\} = 2\varepsilon^{1/N} \sin\left[\frac{2(k-1)\pi}{2N}\right], \qquad k=1,2,\ldots,N \tag{6-3}$$

where $\varepsilon = \sqrt{10^{A_p/10} - 1}$ and A_p is the ripple width in decibels. We set A_p = 3.01 dB or $\varepsilon = 1$ for the Butterworth response. Thus the desired LPP is shown in Fig. 6.3 which is both impedance and frequency normalized. The LPP transfer function is

$$\frac{V_o(s)}{V_{in}(s)} = \frac{1}{2}\left[\frac{1}{s+1}\right] \tag{6-4}$$

which exhibits a dc gain of 1/2 and a pole at $\omega = 1$ radian/sec.

The LPP-to-BR transformation of Eq. 6-2 frequency denormalizes the filter. Using the specified ω_o and B,

$$s = \frac{p(2\pi \text{ x } 200)}{p^2 + (2\pi \text{ x } 3000)^2} \tag{6-5}$$

Examining the effect of the transformation on C_1, we find that a capacitor transforms to a series LC combination as shown in Fig. 6.4(a). In the band reject domain, the ladder becomes as shown in Fig. 6.4(b) which is now frequency denormalized but still impedance normalized. Note that this circuit provides (ideally) infinite rejection at ω_o since the impedance of the L'C' combination is zero at resonance. To provide finite rejection we add R_f in series with L' and C' as shown in Fig. 6.4(c). The transfer function of the circuit of Fig. 6.4(c) is

$$\frac{V_o(s)}{V_{in}(s)} = \frac{1}{2}\left[\frac{s^2 + \left(\frac{R_f}{L'}\right)s + \frac{1}{L'C'}}{s^2 + \left(\frac{R_f + 1/2}{L'}\right)s + \frac{1}{L'C'}}\right] \tag{6-6}$$

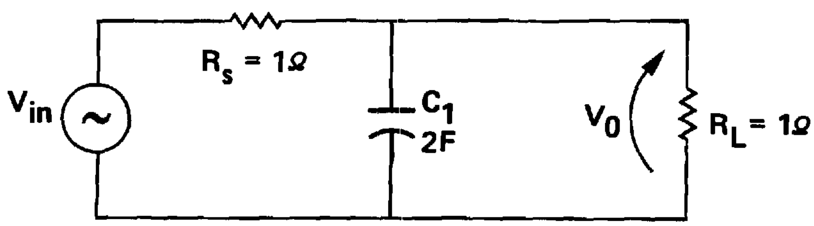


Fig. 6.3 Butterworth low–pass prototype circuit.

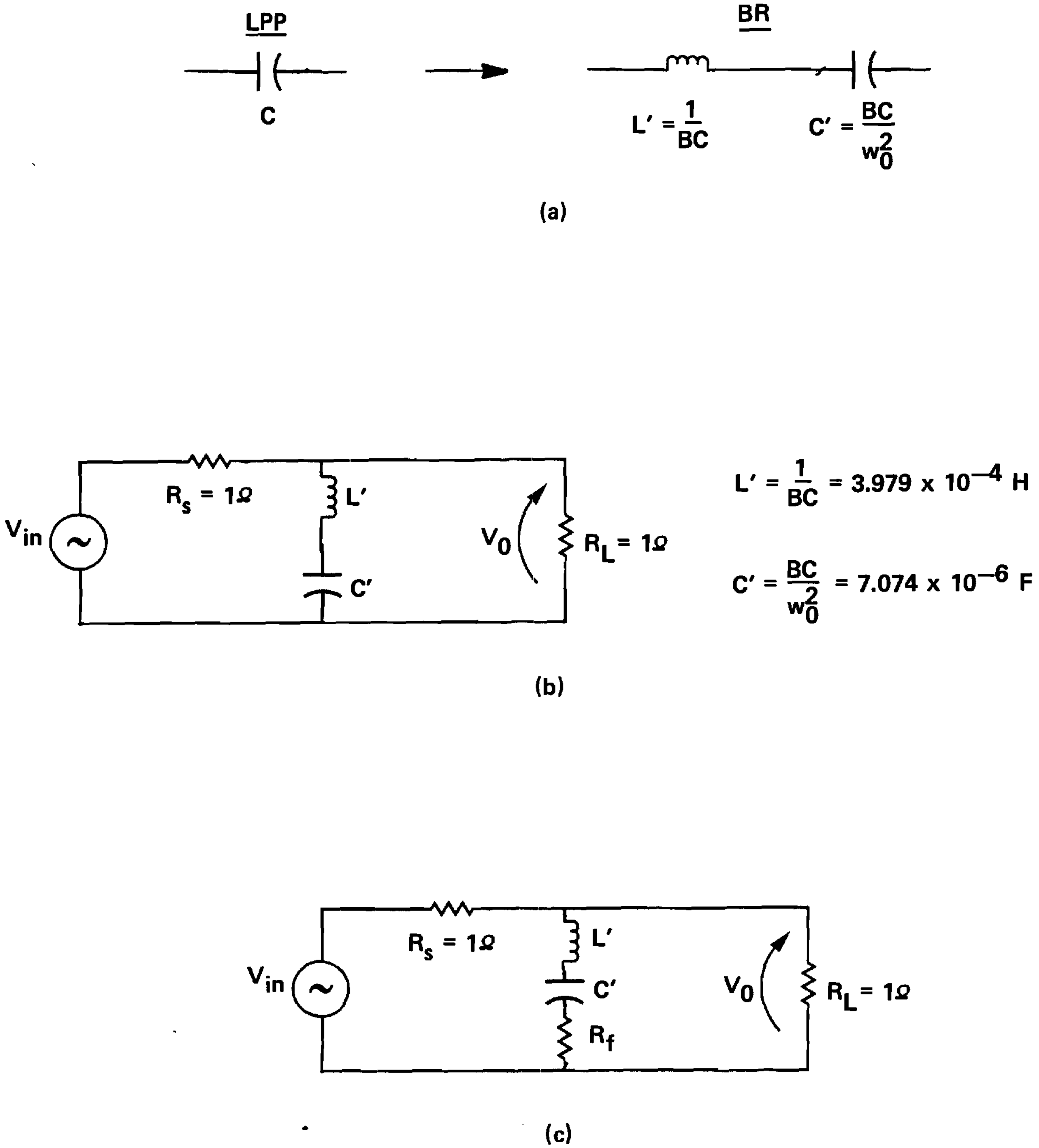


Fig. 6.4 Low–pass to band reject transformation: (a) element substitution; (b) BR circuit with infinite rejection; and (c) BR with finite rejection.

If $R_f \rightarrow 0$ the numerator becomes $(S^2 + \frac{1}{LC})$ and infinite rejection is achieved as expected. Comparing Eqs. 6-1 and 6-6 gives

$$\frac{\omega_o}{Q_n} = \frac{R_f}{L}, \qquad \frac{\omega_o}{Q_d} = \frac{R_f + 1/2}{L} \tag{6-7}$$

The rejection $\frac{Q_n}{Q_d}$ was specified as 30 dB or 31.62. This gives R_f = 0.01633Ω.

Step 4 of our synthesis technique consists of finding a way to convert the ladder of Fig. 6.4(c) to an analog active RC filter. We will use the operational simulation approach [13]. This approach is often referred to as the "leap-frog" method, but the term leap-frog is strictly applicable only to all-pole filters. Since our notch filter includes a pair of transmission zeros, we must use the more generalized signal-flow-graph (SFG) form of operational simulation. The SFG method applies to <u>any</u> LC ladder filter, whether derived using frequency-band transformations or synthesized directly in the frequency domain desired [14].

Operational simulation copies the internal workings of the LC ladder prototype. It produces a circuit in which the signals, say voltages, at various points are the analogs of the voltages and currents of the LC prototype. The active RC structure implements all the current-voltage relationships present in the LC ladder. Hence, the active circuit will consist of <u>integrators</u> that simulate the operation of inductors and capacitors, and <u>summers</u> that simulate the Kirchhoff loop and node equations of the passive LC ladder. Since by this process a one-to-one correspondence is maintained between the integrators in the active circuit and the reactive components in the passive ladder, the resulting active filter retains the low sensitivity properties of the LC prototype [15].

For the moment let us ignore R_f and simulate the ladder of Fig. 6.4(b). We will reintroduce R_f later. Referring to Fig. 6.5, we can write the defining equations

$$I_i = \frac{1}{R_s} (V_{in} - V_o) \tag{6-8}$$

$$I_o = \frac{1}{R_L} (V_o) \tag{6-9}$$

$$I_o = I_i + I_1 \tag{6-10}$$

$$V_o = -I_1 \left(\frac{s^2 \ LC + 1}{sC} \right) \tag{6-11}$$

Note that Eqs. 6-8 and 6-10 define summing operations. Eq. 6-9 is a simple scaling and Eq. 6-11 is a resonance relationship. Now, referring to Eqs. 6-8 through 6-11 and to Fig. 6.6, we construct the SFG representation in Fig. 6.7. Nodes at the top in Fig. 6.7 correspond to signals that simulate ladder currents, while nodes at the bottom represent simulated ladder voltages. The value of a node is defined to be its output.

From Fig. 6.7 it is relatively easy to move to an active RC circuit, since op amps readily implement the operations of summation, scaling, and integration. The two-integrator loop of the SFG is implemented by a 3-amplifier biquad in Fig. 6.8. Although 3 op amps are used here, any biquad circuit would serve. A_1 and A_2 together form a noninverting integrator with time constant R_4C_1. A_3 forms an inverting integrator with time constant R_3C_2. From the standpoint of maximizing the circuits's dynamic range it is advisable to make the two time constants equal [16]. We thus have

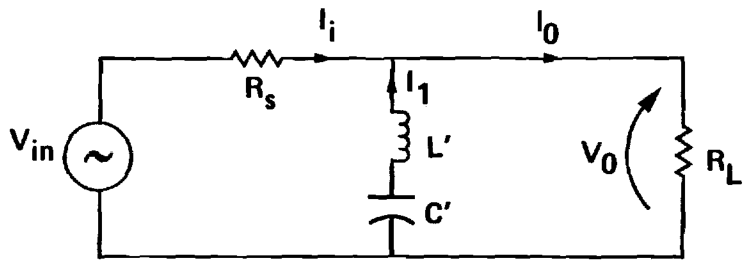


Fig. 6.5 BR circuit with infinite rejection.

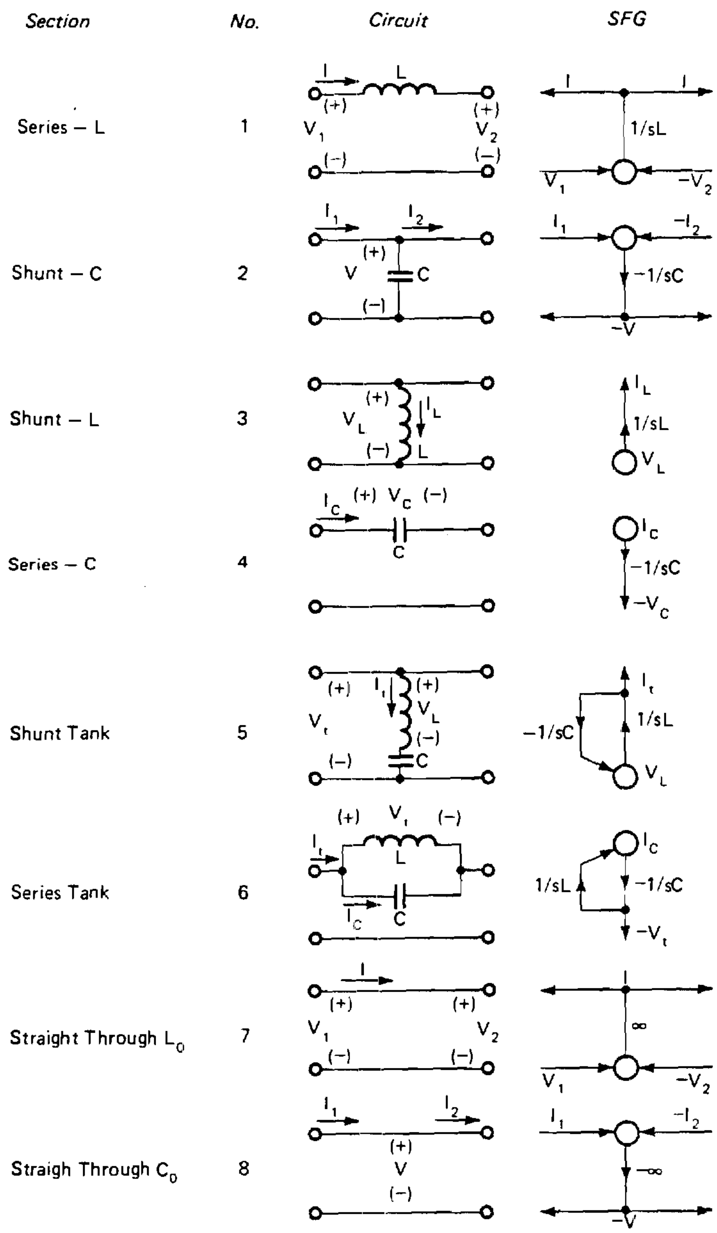


Fig. 6.6 Ladder building blocks and their SFGs (Taken from Sedra and Brackett, pg. 745).

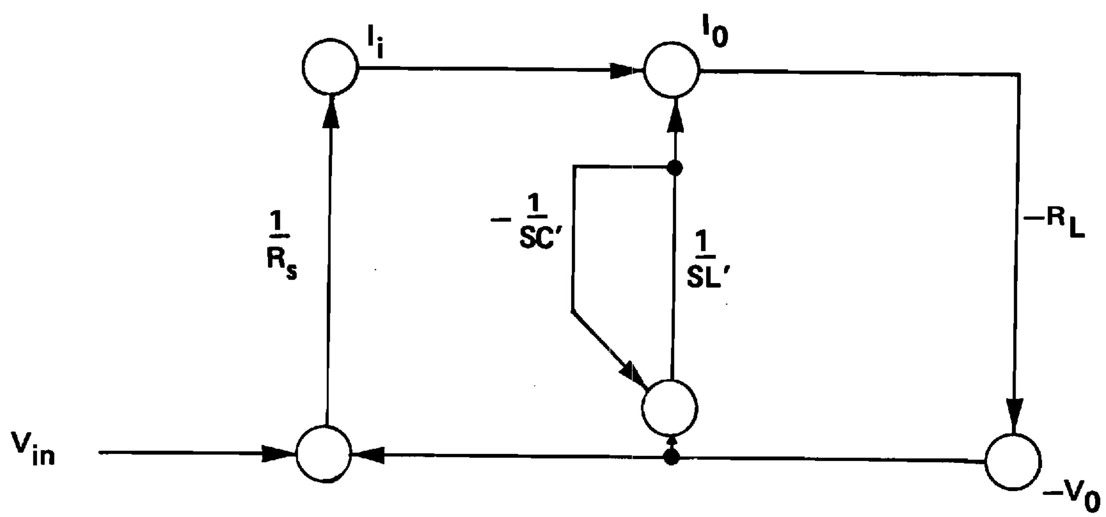


Fig. 6.7 SFG for BR circuit having infinite rejection.

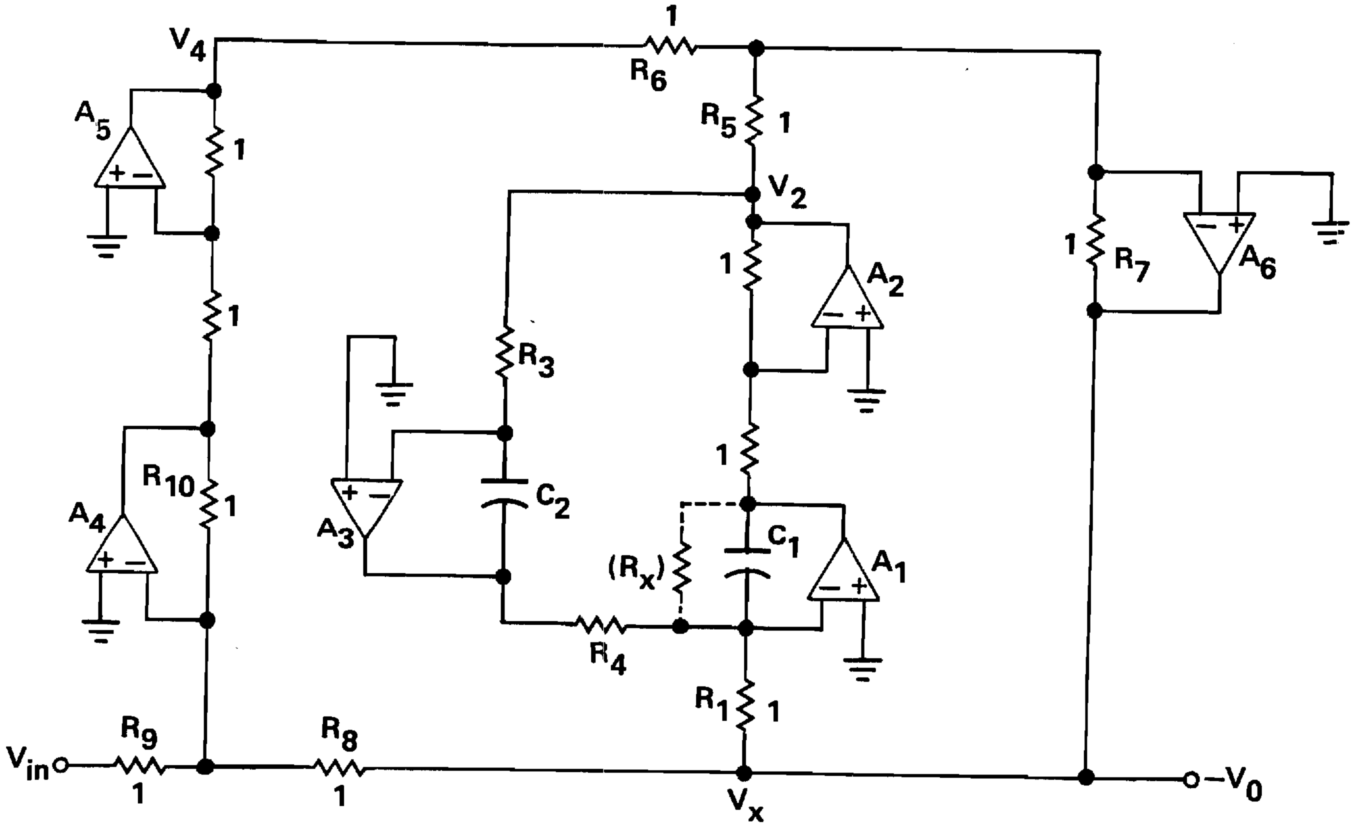


Fig. 6.8 Active RC realization of BR filter.

$$R_4C_1 = R_3C_2 = \sqrt{L'C'} \tag{6-12}$$

Since we had chosen $R_s = R_L = 1\Omega$ in the prototype ladder, all resistors in Fig. 6.8 other than R_3 and R_4 are set equal to 1Ω. However, R_9 will adjust the overall gain of the filter.

The transfer function of the central biquad in Fig. 6.8 is found to be

$$\frac{V_2(s)}{V_x(s)} = \frac{\frac{1}{C_1R_1}\,s}{s^2 + \frac{1}{C_1C_2R_3R_4}} \tag{6-13}$$

with $R_x = \infty$. This is an ideal bandpass function. By embedding the bandpass biquad in the ladder structure we obtain the overall transfer function

$$\frac{V_o(s)}{V_{in}(s)} = -\frac{1}{2}\left[\frac{s^2 + \frac{1}{C_1C_2R_3R_4}}{s^2 + \frac{1}{2C_1R_1}\,s + \frac{1}{C_1C_2R_3R_4}}\right] \tag{6-14}$$

which is a notch function with infinite rejection at $\omega_o = \frac{1}{\sqrt{C_1C_2R_3R_4}}$. To achieve finite rejection we introduce R_x as shown dotted in Fig. 6.8. R_x plays the same role as R_f in Fig. 6.4(c). R_x serves to make A_1 a leaky integrator, i.e. its Q is reduced to a finite value. With R_x in circuit the transfer function is modified to

$$\frac{V_o(s)}{V_{in}(s)} = -\frac{1}{2}\left[\frac{s^2 + \frac{1}{C_1R_x}\,s + \frac{1}{C_1C_2R_3R_4}}{s^2 + \left(\frac{1}{C_1R_x} + \frac{1}{2C_1R_1}\right)s + \frac{1}{C_1C_2R_3R_4}}\right] \tag{6-15}$$

thus achieving finite rejection.

The circuit of Fig. 6.8 represents the culmination of the design work done in steps 1-4. It is a second order notch element resonant at

3000 Hz, implemented in active RC form. It only remains to convert from active RC to switched-capacitor form. An examination of Fig. 6.8 yields only four basic op amp configurations:

(1) inverting summer/amplifier;

(2) noninverting summer/amplifier;

(3) inverting integrator; and

(4) noninverting integrator.

These can be implemented in SC form by using circuits presented in previous chapters. Note that a reduction in the number of op amps will occur in going to SC form. A_1 and A_2 will collapse to a single amplifier, as will A_4 and A_5. The complete SC circuit is shown in Fig. 6.9.

The second order notch example presented here would not normally be implemented as a ladder. The bandpass resonator embedded in the notch ladder is a biquad circuit of the same type as would be used in a conventional cascaded biquad filter. A second order filter requires only one biquad, thus the benefits of multiple interconnections offered by ladder structures do not exist in the second order circuit. But given a sixth order ladder filter, for example, there would be three biquads embedded in the overall ladder structure and the multiple interconnections among the biquads would produce lower sensitivity than a comparable sixth order cascaded biquad filter. Such a sixth order (bandpass) ladder filter is illustrated in Fig. 6.10.

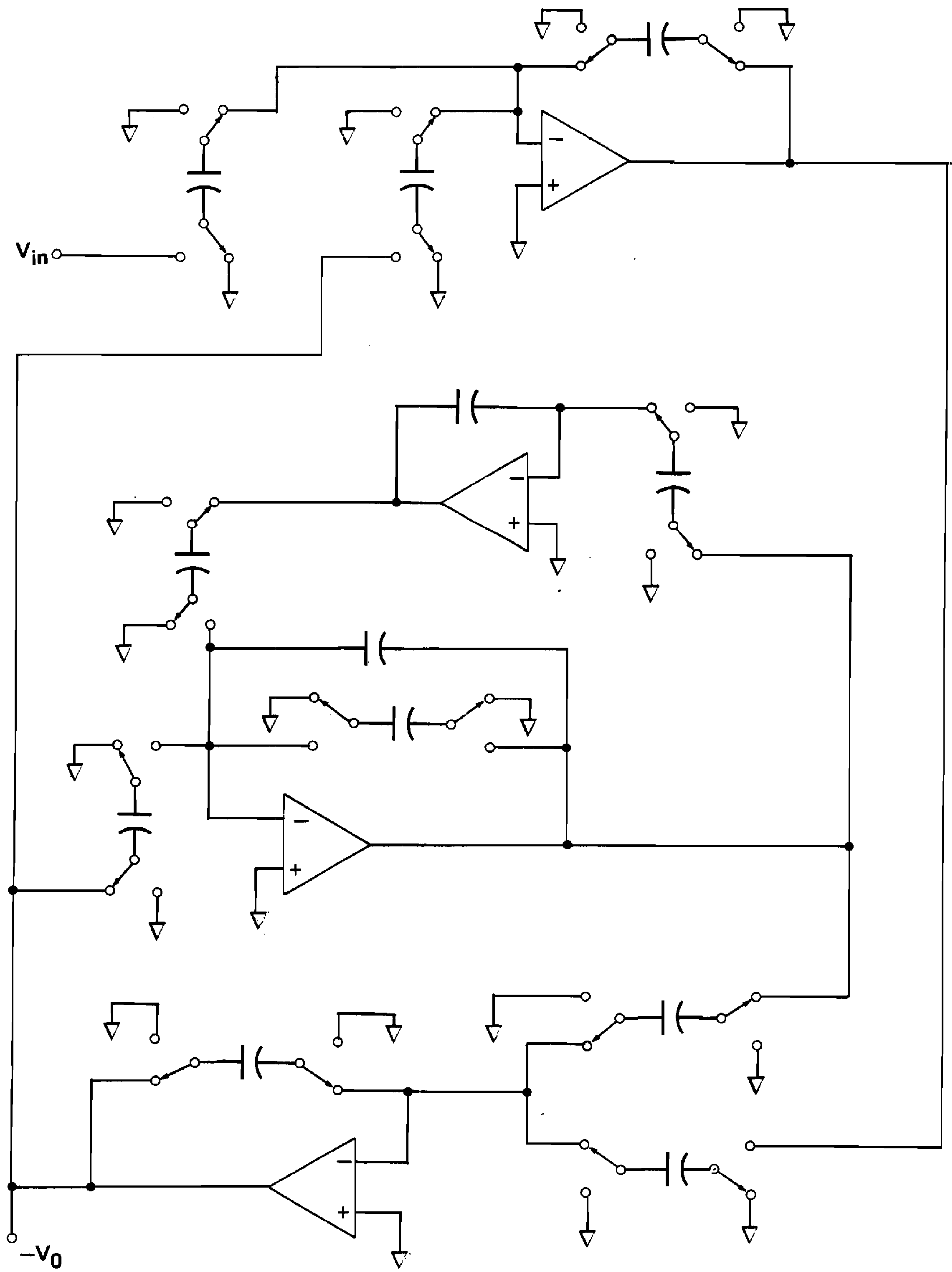


Fig. 6.9 SCF implementation of second order notch filter.

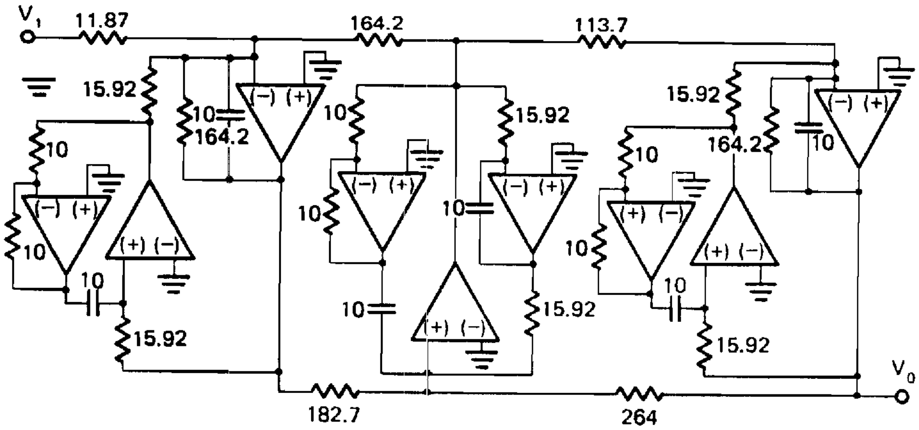


Fig. 6.10 Sixth–order ladder filter for producing a notch response (Taken from Sedra and Bracket, page 730).

REFERENCES

1. G. M. Jacobs, G. F. Landsburg, B. J. White, and D. A. Hodges, "Touch-tone Decoder Chip Mates Analog Filters with Digital Logic," Electronics, Vol. 52, No. 4, February 15, 1979, pp. 105-112.

2. A. Oppenheim and R. Schafer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

3. J. L. McCreary and P. R. Gray, "All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques - Part I," IEEE J. Solid State Circuits, Vol. SC-10, Dec. 1975, pp. 371-379.

4. L. P. Huelsman and P. E. Allen, Introduction to the Theory and Design of Active Filters, McGraw Hill, New York, 1980.

5. R. Gregorian, "Switched-Capacitor Filter Design Using Cascaded Sections," IEEE Trans. Circuits Syst., Vol. CAS-27, June 1980, pp. 515-521.

6. K. Martin and A. S. Sedra, "Exact Design of Switched-Capacitor Band-pass Filter Using Coupled-Biquad Structures," IEEE Trans. Circuits Syst., Vol. CAS-27, June 1980, pp. 469-475.

7. L. P. Huelsman and P. E. Allen, op. cit. pp. 218-221 and 258-262.

8. A. B. Williams, Active Filter Design, Artec House, Inc., Dedham, MA, 1975, pp. 77-80.

9. A. S. Sedra and P. O. Brackett, Filter Theory and Design: Active and Passive, Matrix Publishers, Inc., Portland, OR, 1978, p. 718.

10. Sedra and Brackett, op. cit, p. 142.

11. K. Martin, "Improved Circuits for the Realization of Switched-Capacitor Filters," IEEE Trans. Circuits Systs., Vol. CAS-27, April 1980, pp. 237-244.

12. R. Gregorian, op. cit.

13. Sedra and Brackett, op. cit, Chapter 12.

14. Sedra and Brackett, pg. 731.

15. Sedra and Brackett, pg. 711.

16. Sedra and Brackett, pg. 718.

ACKNOWLEDGEMENTS

The authors wish to thank Honeywell for their support of this project. The efforts of John Timmerman, Bob Payne, and Ken Ceola are greatly appreciated.

APPENDIX A

COMPLETE LISTING FOR MARTLP PROGRAM

```
 OLD MARTLP
*LIST

C       MARTLP
C
C
      DOUBLE PRECISION ALPHA(8),KPRIME,A(4),B(4)
      DOUBLE PRECISION FO,DBGO,GO,FC,T,ALPHAC,ALPHAR,BW,Q
      DOUBLE PRECISION FP,AP,EPS,FREFZ
      DOUBLE PRECISION C(8),CMIN,CSMALL,SUMC,PFPERSQ,CPAREA
      INTEGER STATE,CMD,COUNT,WARPFLG,CAPFLAG,TYPFLAG,MENUFLG
      DO 10 I=1,8
   10 ALPHA(I)=0.
      CMIN=0.
      CSMALL=0.
      SUMC=0.
      FO=0.
      BW=0.
      FC=0.
      T=0.
      KPRIME=0.
      DO 30 I=1,4
      A(I)=0.
   30 B(I)=0.
      GO=0.
      DBGO=0.
      COUNT=0
      WARPFLG=0
      CAPFLAG=0
      PFPERSQ=1.
      CPAREA=0.
      TYPFLAG=0
C
C
C
C
C
      STATE=0
      CALL HEADER
      PRINT, "NEED MENU?  (0=NO, 1=YES)"
      READ, MENUFLG
      IF(MENUFLG.EQ.1) CALL MENU
   90 PRINT,
      PRINT, "ENTER COMMAND NUMBER (#4 FOR MENU)"
      READ , CMD
C
      IERR=0
C
  100 IF(CMD.NE.1) GO TO 200
C
C         TYPFLAG=1 INDICATES BUTTERWORTH FILTER
```

```
C
         CALL TYPHDR(1)
         CALL ENTFO(FO)
         CALL ENTGO(DBGO,GO)
         CALL ENTFC(FC,T)
         CALL ENTALC(ALPHAC)
         CALL ENTALR(ALPHAR)
         CALL BUTTER(FO,BW,Q)
         CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
         STATE=1
         WARPFLG=0
         TYPFLAG=1
 200 IF(CMD.NE.2) GO TO 300
C
C        TYPFLAG=2 INDICATES CHEBYSHEV FILTER
C
         CALL TYPHDR(2)
         CALL ENTFP(FP)
         CALL ENTAP(AP,EPS)
         CALL ENTGO(DBGO,GO)
         CALL ENTFC(FC,T)
         CALL ENTALC(ALPHAC)
         CALL ENTALR(ALPHAR)
         CALL CHEBSHV(FO,FP,BW,AP,EPS,Q)
         CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
         STATE=1
         WARPFLG=0
         TYPFLAG=2
 300 IF(CMD.NE.3) GO TO 400
C
C        TYPFLAG=3 INDICATES GENERALIZED FILTER
C
         CALL TYPHDR(3)
         CALL ENTFO(FO)
         CALL ENTBW(BW)
         CALL ENTGO(DBGO,GO)
         CALL ENTFC(FC,T)
         CALL ENTALC(ALPHAC)
         CALL ENTALR(ALPHAR)
         CALL GENERAL(FO,BW,Q)
         CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
         STATE=1
         TYPFLAG=3
         WARPFLG=0
 400 IF(CMD.NE.4) GO TO 500
         CALL MENU
 500 IF(CMD.NE.5) GO TO 510
         STOP
 510 IF(CMD.NE.6.OR.STATE.EQ.0) GO TO 520
         CALL MWARP(IERR,FP,FO,BW,Q,GO,FC,ALPHA,COUNT,TYPFLAG,
   &AP,EPS,ALPHAR,T,A,B,FREFZ)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=1
 520IF(CMD.NE.7.OR.STATE.LT.1) GO TO 530
```

```
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          IF(CAPFLAG.EQ.1) CALL CAPS(PFPERSQ,CPAREA,ALPHA,C,
     &CSMALL,SUMC,CMIN,ALPHAC)
          CALL PRINT(ALPHA,FO,BW,GO,DBGO,FC,ALPHAC,ALPHAR,T,C,
     &CMIN,SUMC,WARPFLG,CAPFLAG,TYPFLAG,PFPERSQ,CPAREA)
  530 IF(CMD.NE.8.OR.STATE.EQ.0) GO TO 540
          CALL FRESP(ALPHA,T,A,B)
  540 IF(CMD.NE.9.OR.STATE.EQ.0) GO TO 600
          CALL PRECApS(CMIN,ALPHAC,PFPERSQ)
          CALL CAPS(PFPERSQ,CPAREA,ALPHA,C,CSMALL,SUMC,CMIN,ALPHAC)
          CALL POSTCAP(CMIN,ALPHAC,C,SUMC,PFPERSQ,CPAREA)
          CAPFLAG=1
  600 IF(CMD.NE.10.OR.STATE.EQ.0) GO TO 700
          IF(TYPFLAG.EQ.2) PRINT, "NOT VALID FOR CHEBYSHEV"
          IF(TYPFLAG.NE.2) CALL ENTFO(FO)
          IF(TYPFLAG.EQ.1) CALL BUTTER(FO,BW,Q)
          IF(TYPFLAG.EQ.3) CALL GENERAL(FO,BW,Q)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
  700 IF(CMD.NE.11.OR.STATE.EQ.0) GO TO 800
          IF(TYPFLAG.NE.2) PRINT, "NOT VALID FOR NON-CHEBYSHEV"
          IF(TYPFLAG.EQ.2) CALL ENTAP(AP,EPS)
          IF(TYPFLAG.EQ.2) CALL CHEBSHV(FO,FP,BW,AP,EPS,Q)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
  800 IF(CMD.NE.12.OR.STATE.EQ.0) GO TO 900
          IF(TYPFLAG.NE.3) PRINT, "NOT VALID FOR NON-GENERAL"
          IF(TYPFLAG.EQ.3) CALL ENTBW(BW)
          IF(TYPFLAG.EQ.3) CALL GENERAL(FO,BW,Q)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
  900 IF(CMD.NE.13.OR.STATE.EQ.0) GO TO 1000
          CALL ENTFC(FC,T)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
 1000 IF(CMD.NE.14.OR.STATE.EQ.0) GO TO 1100
          CALL ENTALC(ALPHAC)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
 1100 IF(CMD.NE.15.OR.STATE.EQ.0) GO TO 1150
          CALL ENTALR(ALPHAR)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
 1150 IF(CMD.NE.16.OR.STATE.EQ.0) GO TO 1160
          CALL ENTGO(DBGO,GO)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
 1160 IF(CMD.NE.17.OR.STATE.EQ.0) GO TO 1200
          IF(TYPFLAG.NE.2) PRINT, "NOT VALID FOR NON-CHEBYSHEV"
          IF(TYPFLAG.EQ.2) CALL ENTFP(FP)
          IF(TYPFLAG.EQ.2) CALL CHEBSHV(FO,FP,BW,AP,EPS,Q)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
          WARPFLG=0
```

```
100 IF(F.GT.FEND) GO TO 999
    THETA=TH*F
    CALL MAGPHAS(AK,A,B,THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
    PRINT 120, F,DBMAG,ABMAG,DEGPHAS
120 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
    F=F+FINC
    GO TO 100
999 RETURN
    END
C
C
C
C
C
    SUBROUTINE BUTTER(FO,BW,Q)
    DOUBLE PRECISION FO,BW,Q,DSQRT
    BW=DSQRT(2.0D0)*FO
    Q=FO/BW
    RETURN
    END
C
C
    SUBROUTINE CHEBSHV(FO,FP,BW,AP,EPS,Q)
    DOUBLE PRECISION FO,FP,BW,AP,EPS,Q,REPS,ARG1,AOMEGA
    DOUBLE PRECISION DSQRT,DLOG,DCOS,DATAN,DCOSH,DSINH
    EPS=DSQRT(10.0D0**(AP/10.0D0)-1.0D0)
    REPS=1./EPS
    ARG1=DLOG(REPS+DSQRT(REPS**2+1.0D0))/2.
    Q=1./(2.*DCOS(DATAN(DCOSH(ARG1)/DSINH(ARG1))))
    AOMEGA=DSQRT(DSINH(ARG1)**2+0.5D0)
    FO=FP*AOMEGA
    BW=FO/Q
    RETURN
    END

C
C
C
C
C
    SUBROUTINE GENERAL(FO,BW,Q)
    DOUBLE PRECISION FO,BW,Q
    Q=FO/BW
    RETURN
    END
C
C
C
C
C
```

```
1200 IF(CMD.NE.18.OR.STATE.EQ.0) GO TO 1250
          CALL SPECTRM(A,B,T)
1250 IF(CMD.NE.19.OR.STATE.EQ.0) GO TO 1255
          CALL SWEEP(IERR,ALPHA,GO,TYPFLAG,ALPHAR)
1255 IF(CMD.NE.20.OR.STATE.EQ.0) GO TO 1260
          CALL OPTIMIZ(CAPFLAG,ALPHAR,FO,BW,Q,GO,FC,T,
    &ALPHA,A,B,PFPERSQ,CPAREA,C,CSMALL,SUMC,
    &CMIN,ALPHAC,IERR)
          CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
1260 IF(STATE.NE.0) GO TO 1275
     PRINT, "YOUR FIRST COMMAND MUST BE #1, #2, OR #3."
1275 IF(IERR.EQ.0) GO TO 1300
          CALL ERRMSG
1300 GO TO 90
     END
C
C
C
C
     SUBROUTINE ERRMSG
     PRINT, "COMPUTATION HAS BEEN HALTED DUE TO THE"
     PRINT, "OCCURRENCE OF AN ERROR.  CHECK CURRENT"
     PRINT, "PARAMETER VALUES AND RESTART WITH COMMAND"
     PRINT, "1,2, OR 3."
     RETURN
     END
C
C
C
C
     SUBROUTINE SPECTRM(A,B,T)
     DOUBLE PRECISION FBEG,FEND,FINC,F,TH,THETA,AK,A(4),B(4)
     DOUBLE PRECISION ABMAG,DBMAG,RADPHAS,DEGPHAS,T
     PRINT, "CALCULATES MAGNITUDE AND PHASE AT A SERIES"
     PRINT, "OF FREQUENCY POINTS.  ENTER STARTING AND ENDING"
     PRINT, "FREQUENCIES (HERTZ), SEPARATED BY A COMMA."
   5 READ, FBEG,FEND
     IF(FBEG.GE.0.0D0.AND.FEND.GT.0.0D0) GO TO 7
     PRINT, "INVALID PARAMETER(S), PLEASE REENTER FBEG AND FEND"
     GO TO 5
   7 PRINT, "ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)"
  10 READ, FINC
     IF(FINC.GT.0.0D0) GO TO 20
     PRINT, "INVALID INCREMENT; PLEASE ENTER A POSITIVE VALUE"
     GO TO 10
  20 PRINT,
     PRINT,
     PRINT 30,
  30 FORMAT(2X,"FREQUENCY (HZ)",4X,"MAGNITUDE (DB)",4X,
    &"MAGNITUDE",8X,"PHASE (DEG)")
     PRINT,
     TH=2.0D0*3.141592653589793*T
     AK=1.0D0
     F=FBEG
```

```
      SUBROUTINE SWEEP(IERR,ALPHA,GO,TYPFLAG,ALPHAR)
      INTEGER TYPFLAG
      DOUBLE PRECISION FCBEG,FCEND,FCINC,TWOPI,K(2),FC,BW
      DOUBLE PRECISION FO,Q,T,MAGLEAV,GO,DSQRT,ALPHA(8)
      DOUBLE PRECISION A(4),B(4),FLEAV,PHLEAV,FPHLEAV
      DOUBLE PRECISION FREFZ,MAGREFZ,ALPHAR,ALPHAA(8)
      DOUBLE PRECISION GOA,ALPHARA
      PRINT, "ILLUSTRATES EFFECT OF SWEEPING CLOCK FREQUENCIES"
      PRINT, "(FC) ON POLE FREQUENCY AND BANDWIDTH, FOR A"
      PRINT, "SERIES OF CLOCK FREQUENCIES.  ALPHAS AND OTHER"
      PRINT, "PARAMETERS ARE NOT ALTERED BY THIS ROUTINE."
      PRINT, "ENTER BEGINNING AND ENDING CLOCK FREQUENCIES"
      PRINT, "(HERTZ), SEPARATED BY A COMMA."
   10 READ, FCBEG, FCEND
      IF(FCBEG.GT.0.0D0.AND.FCEND.GT.FCBEG) GO TO 20
        PRINT, "ENTER BEGINNING>0 AND END>BEGINNING."
        GO TO 10
   20 PRINT, "ENTER CLOCK FREQUENCY INCREMENT (HERTZ)."
   30 READ, FCINC
      IF(FCINC.GT.0.0D0) GO TO 40
        PRINT, "ENTER INCREMENT>0."
        GO TO 30
   40 PRINT,
      PRINT,
      PRINT 50,
   50 FORMAT(2X,"CLOCK",15X,"FO",16X,"BW",13X,"PASSBAND")
      PRINT 60,
   60 FORMAT(2X,"FREQ.",50X,"EDGE")
   90 TWOPI=2.0D0*3.141592653589793
      K(1)=ALPHA(3)/TWOPI
      K(2)=ALPHA(4)*DSQRT(ALPHAR)/TWOPI
C
      DO 300 I=1,8
  300 ALPHAA(I)=ALPHA(I)
      GOA=GO
      ALPHARA=ALPHAR
C
C
      FC=FCBEG
  100 IF(FC.GT.FCEND) GO TO 999
      BW=K(1)*FC
      FO=K(2)*FC
      Q=FO/BW
      T=1.0D0/FC
      IF(TYPFLAG.EQ.1) MAGLEAV=GO/DSQRT(2.0D0)
      IF(TYPFLAG.EQ.2) MAGLEAV=GO
      IF(TYPFLAG.EQ.3) MAGLEAV=(Q*ALPHA(4)*ALPHA(7)*GO/(T**2))
     &/((TWOPI*FO)**2)
      PHLEAV=-90.0D0
      ICOUNT=0
      ICHOICE=2
C
C
C
```

```
      CALL FINDPK(IERR,FO,Q,GOA,FC,T,ALPHAA,A,B,MAGLEAV,
     &FLEAV,PHLEAV,FPHLEAV,FREFZ,MAGREFZ,TYPFLAG,
     &ICOUNT,ICHOICE,ALPHARA,BW)
      IF(IERR.EQ.1) RETURN
  190 PRINT 200, FC,FO,BW,FLEAV
  200 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
C
C
      FC=FC+FCINC
      GO TO 100
  999 RETURN
      END
C
C
C
C
C
      SUBROUTINE OPTIMIZ(CAPFLAG,ALPHAR,FO,BW,Q,GO,FC,
     &T,ALPHA,A,B,PFPERSQ,CPAREA,C,CSMALL,SUMC,
     &CMIN,ALPHAC,IERR)
      INTEGER CAPFLAG
      DOUBLE PRECISION PARAM,PMIN,PMAX,PINC,PBEST,SUMNEW,SUMBEST
      DOUBLE PRECISION ALPHAR,FO,BW,Q,GO,FC,T,ALPHA(8)
      DOUBLE PRECISION A(4),B(4),PFPERSQ,CPAREA,C(8)
      DOUBLE PRECISION CSMALL,SUMC,CMIN,ALPHAC
C
      IF(CAPFLAG.EQ.1) GO TO 50
        PRINT, "PLEASE CALCULATE CAPACITORS (COMMAND #9) FIRST."
        RETURN
   50 PRINT, "TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE"
      PRINT, "PARAMETER FROM THE FOLLOWING LIST:"
      PRINT, "      1.  ALPHAC=C2/C1"
      PRINT, "      2.  ALPHAR=ALPHA(5)/ALPHA(2)"
      READ, ICHOICE
      IF(ICHOICE.NE.1.AND.ICHOICE.NE.2) GO TO 50
      PRINT,
   60 PRINT, "ENTER IN ORDER, SEPARATED BY COMMAS:"
      PRINT, "PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT"
      READ, PMIN, PMAX, PINC
      IF(PMIN.GT.0.0D0.AND.PMAX.GT.PMIN.AND.PINC.GT.0.0D0)
     &GO TO 70
      PRINT, "ENTER PMAX>PMIN>0 AND INCREMENT>0."
      GO TO 60
C
C
   70 PARAM=PMIN
      PBEST=PARAM
      SUMBEST=SUMC*1.0D6
C
   80 IF(PARAM.GT.PMAX) GO TO 200
      IF(ICHOICE.EQ.1) ALPHAC=PARAM
      IF(ICHOICE.EQ.2) ALPHAR=PARAM
      CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,GO,FC,T,ALPHA,A,B)
      IF(IERR.EQ.1) RETURN
      CALL CAPS(PFPERSQ,CPAREA,ALPHA,C,CSMALL,SUMC,
     &CMIN,ALPHAC)
```

```
C
C
C
      SUMNEW=SUMC
      IF(SUMNEW.GE.SUMBEST) GO TO 100
        PBEST=PARAM
        SUMBEST=SUMNEW
C
  100 CONTINUE
      PARAM=PARAM+PINC
C
C
      GO TO 80
C
C
  200 CONTINUE
      PRINT, "PARAMETER NAME:"
      IF(ICHOICE.EQ.1) PRINT, "ALPHAC"
      IF(ICHOICE.EQ.2) PRINT, "ALPHAR"
      PRINT 210, PBEST
  210 FORMAT("FOR PARAMETER= ",F15.8,"  ,")
      PRINT 220, SUMBEST
  220 FORMAT("THE SMALLEST VALUE OF TOTAL CAP. = ",G15.8,"  PF")
C
      IF(ICHOICE.EQ.1) ALPHAC=PBEST
      IF(ICHOICE.EQ.2) ALPHAR=PBEST
C
      PRINT,
      PRINT,
      PRINT, "THE OPTIMIZED PARAMETER HAS BEEN PLACED"
      PRINT, "IN THE ARGUMENT LIST."
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE MAGPHAS(AK,A,B,THETA,ABMG,DBMG,RADPHAS,DEGPHAS)
      DOUBLE PRECISION AK,A(4),B(4),THETA,ABMG,DBMG,RADPHAS
      DOUBLE PRECISION DEGPHAS,PI,COSTH,COS2TH,COS3TH,SINTH
      DOUBLE PRECISION SIN2TH,SIN3TH,AMAGNUM,ARGNUM,AKPHASE
      DOUBLE PRECISION AMAGDEN,ARGDEN,RENUM,AIMNUM,REDEN,AIMDEN
      DOUBLE PRECISION DABS,DCOS,DSIN,DSQRT,DLOG10,DATAN2
      PI=3.141592653589793
C
   10 IF(THETA.LT.1.0D3) GO TO 20
          THETA=THETA-PI
          GO TO 10
C
   20 COSTH=DCOS(THETA)
      COS2TH=DCOS(2.0D0*THETA)
      COS3TH=DCOS(3.0D0*THETA)
      SINTH=DSIN(THETA)
      SIN2TH=DSIN(2.0D0*THETA)
      SIN3TH=DSIN(3.0D0*THETA)
      AKPHASE=PI
```

```
      IF(AK.GE.0.0D0) AKPHASE=0.0D0
      RENUM=A(4)*COS3TH+A(3)*COS2TH+A(2)*COSTH+A(1)
      AIMNUM=A(4)*SIN3TH+A(3)*SIN2TH+A(2)*SINTH
      REDEN=B(4)*COS3TH+B(3)*COS2TH+B(2)*COSTH+B(1)
      AIMDEN=B(4)*SIN3TH+B(3)*SIN2TH+B(2)*SINTH
      AMAGNUM=DSQRT(RENUM*RENUM+AIMNUM*AIMNUM)
      AMAGDEN=DSQRT(REDEN*REDEN+AIMDEN*AIMDEN)
      ABMG=(AMAGNUM/AMAGDEN)*DABS(AK)
      RADPHAS=DATAN2(AIMNUM,RENUM)-DATAN2(AIMDEN,REDEN)+AKPHASE
      IF(ABMG.LT.0.0D0) PRINT, "ABMG= ",ABMG
      DBMG=20.*DLOG10(ABMG)
      DEGPHAS=RADPHAS*180./PI
      RETURN
      END
C
C
C
C
C
      SUBROUTINE FRESP(ALPHA,T,A,B)
      DOUBLE PRECISION ALPHA(8),KPRIME,A(4),B(4),ABMAG
      DOUBLE PRECISION PI,T,F,THETA,DBMAG,DEGPHAS,RADPHAS
      PI=3.141592653589793
      PRINT, "ENTER EVALUATION FREQUENCY IN HERTZ"
      READ, F
      KPRIME=1.
      THETA=2*PI*F*T
C
      DBMAG=0.
      DEGPHAS=0.
      CALL MAGPHAS(KPRIME,A,B,THETA,ABMAG,DBMAG,
     &RADPHAS,DEGPHAS)
C
      PRINT 10, F
   10 FORMAT("AT F= ",G15.8,"  HERTZ")
      PRINT 20, DBMAG
   20 FORMAT(" MAGNITUDE= ",G15.8,"  DB")
      PRINT 30, DEGPHAS
   30 FORMAT(" PHASE= ",G15.8,"  DEGREES")
C
C
C
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ALPHAS(IERR,ALPHAR,F0,BW,Q,G0,FC,T,ALPHA,A,B)
      DOUBLE PRECISION ALPHA(8),A(4),B(4),ALPHAR,F0,BW,Q
      DOUBLE PRECISION G0,FC,T,TWOPI
      TWOPI=2.*3.141592653589793
      T=1./FC
C
```

```
C
      BW=F0/Q
      ALPHA(3)=TWOPI*BW*T
      ALPHA(4)=(TWOPI*F0*T)/DSQRT(ALPHAR)
      ALPHA(7)=ALPHAR*ALPHA(4)
      ALPHA(5)=ALPHA(7)*G0
      ALPHA(6)=0.0D0
C
      A(4)=0.
      A(3)=0.
      A(2)=ALPHA(4)*ALPHA(5)
      A(1)=0.
      B(4)=0.
      B(3)=1.
      B(2)=ALPHA(4)*ALPHA(7)+ALPHA(3)-2.
      B(1)=1.-ALPHA(3)
C
C
      DO 400 I=3,7
      IF(ALPHA(I).GE.0.0D0) GO TO 400
      PRINT, "WARNING:  NEGATIVE ALPHA VALUE--"
      PRINT, "RECHECK INPUT PARAMETERS."
      PRINT 410, I, ALPHA(I)
  410 FORMAT("ALPHA(",I1,") = ",G15.8)
      IERR=1
  400 CONTINUE
      IF(IERR.EQ.1) RETURN
C
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PRECAPS(CMIN,ALPHAC,PFPERSQ)
      DOUBLE PRECISION ALPHAC,CMIN,PFPERSQ
          PRINT, "ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF"
          READ, CMIN
      PRINT, "ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL"
      READ, PFPERSQ
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CAPS(PFPERSQ,CPAREA,ALPHA,C,CSMALL,SUMC,CMIN,ALPHAC)
      DOUBLE PRECISION ALPHA(8),C(8),PFPERSQ,CPAREA,CSMALL
      DOUBLE PRECISION SUMC,CMIN,ALPHAC
C
C
      C(1)=1.
      C(2)=ALPHAC*C(1)
      C(4)=ALPHA(4)*C(2)
      C(3)=ALPHA(3)*C(1)
      C(5)=ALPHA(5)*C(1)
      C(7)=ALPHA(7)*C(1)
```

```
C
      CSMALL=AMIN1(C(1),C(2),C(3),C(4),C(5),C(7))
      DO 10 I=1,5
   10 C(I)=C(I)*CMIN/CSMALL
      C(7)=C(7)*CMIN/CSMALL
      SUMC=0.
      DO 20 I=1,5
   20 SUMC=SUMC+C(I)
      SUMC=SUMC+C(7)
      CPAREA=SUMC/PFPERSQ
C
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE POSTCAP(CMIN,ALPHAC,C,SUMC,PFPERSQ,CPAREA)
      DOUBLE PRECISION C(8),CMIN,ALPHAC,SUMC,PFPERSQ,CPAREA
          PRINT 340, CMIN
  340     FORMAT("FOR MINIMUM C= ",G15.8,"  PF")
          PRINT 350, ALPHAC
  350     FORMAT("AND C2/C1= ",F15.8," ,THE CAPACITANCE VALUES ARE:")
          PRINT,
          DO 360 I=1,5
          PRINT 370, I,C(I)
  370     FORMAT("C(",I1,")= ",G15.8,"  PF")
  360     CONTINUE
      PRINT 372, C(7)
  372 FORMAT("C(7)= ",G15.8,"  PF")
          PRINT,
          PRINT 380, SUMC
  380     FORMAT("TOTAL CAPACITANCE USED= ",G15.8,"  PF")
      PRINT 390, PFPERSQ,CPAREA
  390 FORMAT("AT ",G11.4," PF/SQ, AREA USED = ",G13.6,"  SQ MILS")
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PRINT(ALPHA,FO,BW,GO,DBGO,FC,ALPHAC,ALPHAR,T,C,
     &CMIN,SUMC,WARPFLG,CAPFLAG,TYPFLAG,PFPERSQ,CPAREA)
      DOUBLE PRECISION ALPHA(8),C(8),FO,BW,GO,DBGO,FC,ALPHAC
      DOUBLE PRECISION ALPHAR,T,CMIN,SUMC,PFPERSQ,CPAREA
      INTEGER CAPFLAG,WARPFLG,TYPFLAG
      CALL TYPHDR(TYPFLAG)
      PRINT, "VALUES AS CURRENTLY CALCULATED ARE:"
      PRINT 100, FO
  100 FORMAT("FO= ",G15.8,"  HERTZ")
      PRINT 110, BW
  110 FORMAT("BW= ",G15.8,"  HERTZ")
```

```
      PRINT 120, GO
  120 FORMAT("GO= ",G15.8,"  (DIMENSIONLESS)")
      PRINT 130, DBGO
  130 FORMAT("DBGO= ",G15.8,"  DECIBELS")
      PRINT 140, FC
  140 FORMAT("FC= ",G15.8,"  HERTZ")
      PRINT 150, T
  150 FORMAT("T= ",G15.8,"  SECONDS")
      PRINT 160, ALPHAC
  160 FORMAT("ALPHAC=ALPHA(2)/ALPHA(1)= ",F15.8)
      PRINT 170, ALPHAR
  170 FORMAT("ALPHAR=ALPHA(7)/ALPHA(4)= ",F15.8)
    4 DO 10 I=3,5
      PRINT 5, I,ALPHA(I)
    5 FORMAT("ALPHA(",I1,")= ",G15.8)
   10 CONTINUE
      PRINT 11, ALPHA(7)
   11 FORMAT("ALPHA(7)= ",G15.8)
      IF(CAPFLAG.EQ.1) GO TO 12
          GO TO 30
   12 DO 20 I=1,5
      PRINT 15, I,C(I)
   15 FORMAT("C(",I1,")= ",G15.8,"  PF")
   20 CONTINUE
      PRINT 372, C(7)
  372 FORMAT("C(7)= ",G15.8,"  PF")
      PRINT 200, CMIN
  200 FORMAT("MINIMUM C= ",G15.8,"  PF")
      PRINT 210, SUMC
  210 FORMAT("TOTAL C= ",G15.8,"  PF")
      PRINT 390, PFPERSQ,CPAREA
  390 FORMAT("AT ",G11.4," PF/SQ, AREA USED = ",G13.6,"  SQ MILS")
   30 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTFO(FO)
      DOUBLE PRECISION FO
      PRINT, "ENTER POLE FREQUENCY, FO (HERTZ)"
      READ, FO
      RETURN
      END
C
C
C
C
C
```

```
      SUBROUTINE ENTFP(FP)
      DOUBLE PRECISION FP
      PRINT, "ENTER FP= FREQUENCY WHERE MAGNITUDE "
      PRINT, "CHARACTERISTIC LEAVES PASSBAND (HERTZ)."
      READ, FP
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTBW(BW)
      DOUBLE PRECISION BW
      PRINT, "ENTER BANDWIDTH, BW (HERTZ)"
      READ, BW
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTGO(DBGO,GO)
      DOUBLE PRECISION DBGO,GO
      PRINT, "ENTER GAIN AT F=0, DBGO (DECIBELS)"
      READ, DBGO
      GO=10.**(DBGO/20.)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTFC(FC,T)
      DOUBLE PRECISION FC,T
      PRINT, "ENTER CLOCK FREQUENCY, FC (HERTZ)"
      READ, FC
      T=1./FC
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTALC(ALPHAC)
      DOUBLE PRECISION ALPHAC
      PRINT, "ENTER ALPHAC=C2/C1"
      READ, ALPHAC
      RETURN
      END
C
```

```
C
C
C
C
      SUBROUTINE ENTALR(ALPHAR)
      DOUBLE PRECISION ALPHAR
      PRINT, "ENTER ALPHAR=ALPHA(7)/ALPHA(4)"
      READ, ALPHAR
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTAP(AP,EPS)
      DOUBLE PRECISION AP,EPS,DSQRT
      PRINT, "ENTER RIPPLE WIDTH, AP  (DECIBELS)"
      READ, AP
      EPS=DSQRT(10.0D0**(AP/10.0D0)-1.0D0)
      RETURN
      END
C
C
C
C
      SUBROUTINE HEADER
      PRINT, "SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR"
      PRINT, "MARTIN LOWPASS FILTER."
      PRINT, "FORTRAN VERSION BY W.N. GOOLSBY."
      PRINT, "REVISION 6/5/81."
      PRINT,
      RETURN
      END
C
C
C
C
C
      SUBROUTINE TYPHDR(TYPFLAG)
      INTEGER TYPFLAG
      IF(TYPFLAG.EQ.1) PRINT, "FILTER TYPE:  BUTTERWORTH"
      IF(TYPFLAG.EQ.2) PRINT, "FILTER TYPE: CHEBYSHEV"
      IF(TYPFLAG.EQ.3) PRINT, "FILTER TYPE: GENERALIZED"
      RETURN
      END
C
C
C
C
```

```
      SUBROUTINE MENU
      PRINT, "A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,"
      PRINT, "ENTER THE NUMBER OF THE APPROPRIATE COMMAND."
      PRINT,
      PRINT, "1.  ENTER PARAMETERS FOR BUTTERWORTH LOWPASS FILTER"
      PRINT, "2.  ENTER PARAMETERS FOR CHEBYSHEV LOWPASS FILTER"
      PRINT, "3.  ENTER PARAMETERS FOR GENERALIZED LOWPASS FILTER"
      PRINT, "4.  PRINT MENU"
      PRINT, "5.  TERMINATE PROGRAM"
      PRINT, "6.  WARP FO, BW, AND GAIN."
      PRINT, "7.  PRINT CURRENT VALUES."
      PRINT, "8.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F"
      PRINT, "9.  FIND CAPACITOR VALUES AND TOTAL C"
      PRINT, "10.  CHANGE FO ONLY"
      PRINT, "11.  CHANGE CHEBYSHEV RIPPLE WIDTH ONLY"
      PRINT, "12.  CHANGE GENERALIZED FILTER BANDWIDTH ONLY"
      PRINT, "13.  CHANGE FC ONLY"
      PRINT, "14.  CHANGE ALPHAC=C(2)/C(1) ONLY"
      PRINT, "15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4)  ONLY"
      PRINT, "16.  CHANGE GAIN AT F=0 ONLY (DECIBELS)"
      PRINT, "17.  CHANGE CHEBYSHEV PASSBAND EDGE, FP, ONLY."
      PRINT, "18.  FIND MAGNITUDE AND PHASE AT A SERIES"
      PRINT, "     OF FREQUENCY POINTS."
      PRINT, "19.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON FO & BW"
      PRINT, "20.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER"
      RETURN
      END
C
C
C
C
C
      SUBROUTINE WARPBG(IERR,FO,BW,Q,GO,FC,ALPHA,COUNT,
     &COUNT2,CHOICE,TYPFLAG,T,A,B,ALPHAR)
      DOUBLE PRECISION MAGLEAV,ALPHA(8),MAGDESO,MAGFRAC
      DOUBLE PRECISION A(4),B(4),MAGERR,MAGREFZ,PI,FO,BW,Q
      DOUBLE PRECISION GO,FC,T,ALPHAR,FDES,PHASDES,QDES
      DOUBLE PRECISION AK,FERROR,FFRAC,PHLEAV,FLEAV,FPHLEAV
      DOUBLE PRECISION FREFZ,FFOUND,FDEVN,AFDEVN,GDEVN
      DOUBLE PRECISION AGDEVN,DSQRT,DABS
      INTEGER COUNT,COUNT2,RUNFLAG,MAXRUNS,CHOICE,TYPFLAG
      PI=3.141592653589793
      FDES=FO
      PHASDES=-90.0
      MAGDESO=GO
      QDES=FO/BW
      AK=ALPHA(4)*ALPHA(7)*GO/(T**2)
      IF(TYPFLAG.EQ.1) MAGLEAV=GO/DSQRT(2.0D0)
      IF(TYPFLAG.EQ.3) MAGLEAV=AK*QDES/((2.*PI*FDES)**2)
      FERROR=FDES*1.0E-5
      MAGERR=MAGDESO*1.0E-4
      FFRAC=1.0
      MAGFRAC=1.0
      MAXRUNS=30
      COUNT=0
      COUNT2=0
C
```

```
      Q=QDES
      PHLEAV=PHASDES
   10 CALL FINDPK(IERR,FO,Q,GO,FC,T,ALPHA,A,B,MAGLEAV,FLEAV,
     &PHLEAV,FPHLEAV,FREFZ,MAGREFZ,TYPFLAG,
     &COUNT2,CHOICE,ALPHAR,BW)
      IF(IERR.EQ.1) RETURN
      RUNFLAG=0
      COUNT=COUNT+1
      IF(CHOICE.EQ.1) FFOUND=FLEAV
      IF(CHOICE.EQ.2) FFOUND=FPHLEAV
      FDEVN=FDES-FFOUND
      AFDEVN=ABS(FDEVN)
      IF(AFDEVN.LT.FERROR) GO TO 20
          FO=FO+FFRAC*FDEVN
          IF(TYPFLAG.EQ.1) CALL BUTTER(FO,BW,Q)
          IF(TYPFLAG.EQ.3) CALL GENERAL(FO,BW,Q)
         RUNFLAG=1
   20 GDEVN=MAGDESO-GO
      AGDEVN=ABS(GDEVN)
      IF(AGDEVN.LT.MAGERR) GO TO 30
          GO=GO+MAGFRAC*GDEVN
          RUNFLAG=1
   30 IF(RUNFLAG.EQ.1) GO TO 40
          RETURN
   40 IF(COUNT.LE.MAXRUNS) GO TO 10
      PRINT, "MAXRUNS EXCEEDED IN WARPBG."
      PRINT, "MAXIMUM ITERATION COUNT EXCEEDED WHILE"
      PRINT, "TRYING TO WARP."
      IERR=1
          RETURN
          END
C
C
C
C
C
      SUBROUTINE WARPCHB(IERR,FP,AP,FO,BW,Q,GO,FC,ALPHA,COUNT,COUNT2,
     &T,A,B,EPS,ALPHAR,FREFZ)
      INTEGER COUNT,COUNT2,RUNFLAG,MAXRUNS,TYPFLAG,CHOICE
      DOUBLE PRECISION MAGLEAV,MAGDESO,MAGERR,MAGFRAC
      DOUBLE PRECISION MAGREFZ,A(4),B(4),FP,AP,FO,BW,Q,GO
      DOUBLE PRECISION FC,ALPHA(8),T,ALPHAR,FREFZ,PI,FDES
      DOUBLE PRECISION  RIPDES,RIPERR,RIPFRAC,FERROR,FFRAC
      DOUBLE PRECISION PHLEAV,FLEAV,FPHLEAV,EXPRIP,RIPDEVN
      DOUBLE PRECISION ARIPDEV,EPS,GDEVN,AGDEVN,FDEVN,AFDEVN
      DOUBLE PRECISION DABS,DLOG10
      PI=3.141592653589793
      FDES=FP
      MAGDESO=GO
      MAGLEAV=GO
      RIPDES=AP
      MAGERR=MAGDESO*1.0E-5
      MAGFRAC=1.0
      RIPERR=RIPDES*1.0E-5
      RIPFRAC=1.0
```

```
      FERROR=FDES*1.0E-5
      FFRAC=1.0
      PHLEAV=-90.0
      MAXRUNS=50
      COUNT=0
      COUNT2=0
      TYPFLAG=2
      CHOICE=1
C
C
C
C
   10 CALL FINDPK(IERR,FO,Q,GO,FC,T,ALPHA,A,B,
     &MAGLEAV,FLEAV,PHLEAV,FPHLEAV,FREFZ,
     &MAGREFZ,TYPFLAG,COUNT2,CHOICE,ALPHAR,BW)
      IF(IERR.EQ.1) RETURN
      RUNFLAG=0
      COUNT=COUNT+1
      EXPRIP=20.*DLOG10(MAGREFZ/GO)
      RIPDEVN=RIPDES-EXPRIP
      ARIPDEV=ABS(RIPDEVN)
      IF(ARIPDEV.LT.RIPERR) GO TO 20
          AP=AP+RIPFRAC*RIPDEVN
          CALL CHEBSHV(FO,FDES,BW,AP,EPS,Q)
          RUNFLAG=1
   20 GDEVN=MAGDESO-GO
      AGDEVN=ABS(GDEVN)
      IF(AGDEVN.LT.MAGERR) GO TO 25
          GO=GO+MAGFRAC*GDEVN
          RUNFLAG=1
   25 FDEVN=FDES-FLEAV
      AFDEVN=ABS(FDEVN)
      IF(AFDEVN.LT.FERROR.AND.RUNFLAG.EQ.0) GO TO 30
          FP=FP+FFRAC*FDEVN
          CALL CHEBSHV(FO,FP,BW,AP,EPS,Q)
          RUNFLAG=1
   30 IF(RUNFLAG.EQ.1) GO TO 40
          GO TO 100
   40 IF(COUNT.LE.MAXRUNS) GO TO 10
          PRINT, "MAXRUNS EXCEEDED IN WARPCHB"
      PRINT, "MAXIMUM ITERATION COUNT EXCEEDED WHILE"
      PRINT, "TRYING TO WARP."
      IERR=1
          RETURN
  100 CONTINUE
C
C
C
      RETURN
      END
C
C
```

```
C
C
      SUBROUTINE FINDPK(IERR,FO,Q,EXPGO,FC,T,ALPHA,A,B,MAGLEAV,FLEAV,
     &PHLEAV,FPHLEAV,FREFZ,MAGREFZ,TYPFLAG,COUNT,CHOICE,ALPHAR,BW)
      DOUBLE PRECISION MAGLEAV,MAGREFZ,A(4),B(4),ALPHA(8)
      DOUBLE PRECISION LASTMAG,LASTPHS,FO,Q,EXPGO,FC,T
      DOUBLE PRECISION FLEAV,PHLEAV,FPHLEAV,FREFZ,ALPHAR,BW
      DOUBLE PRECISION PI,THETA,AK,ABMAG,DBMAG,RADPHAS
      DOUBLE PRECISION DEGPHAS,RROOT2,F,FFR,FLO,FHI,DIR
      DOUBLE PRECISION STEP,ERROR,DSQRT
      INTEGER TYPFLAG,COUNT,CHOICE,CROSSED,PEAKING
      PI=3.141592653589793
      PEAKING=0
C
      CALL ALPHAS(IERR,ALPHAR,FO,BW,Q,EXPGO,FC,T,ALPHA,A,B)
      IF(IERR.EQ.1) RETURN
C
C     FIND EXPGO
C
      THETA=0.
      AK=1.
      CALL MAGPHAS(AK,A,B,THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      EXPGO=ABMAG
C
C     FIND FLEAV=FREQ. WHERE CHARACTERISTIC LEAVES PASSBAND
C
C      RROOT2 IS MADE SLIGHTLY LARGER THAN ACTUAL
C      1/DSQRT(2) TO ALLOW FOR MACHINE IMPRECISION
C      IN COMPUTING "Q".
C
C
      RROOT2=1./DSQRT(2.0D0)+1.0D-7
      IF(Q.GT.RROOT2) PEAKING=1
      IF(PEAKING.EQ.0) GO TO 100
          F=FO
          FFR=DSQRT(1.-1./(2.*Q**2))*FO
          FLO=FFR
          FHI=100.*FLO
          DIR=1.
          STEP=FFR/100.
          ERROR=FO*1.0E-7
          ABMAG=EXPGO*1.0E4
          GO TO 210
  100 F=0.
      FLO=0.
      FHI=20.*FO
      DIR=1.
      STEP=FO/20.
      ERROR=FO*1.0E-7
      ABMAG=EXPGO
  210 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(AK,A,B,THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
```

```
      COUNT=COUNT+1
      CROSSED=0
      IF(ABMAG.LT.MAGLEAV.AND.LASTMAG.GT.MAGLEAV) CROSSED=1
      IF(LASTMAG.LT.MAGLEAV.AND.ABMAG.GT.MAGLEAV) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 220
          DIR=-1.*DIR
          STEP=STEP/2.
  220 F=F+STEP*DIR
      IF(STEP.GE.ERROR) GO TO 230
          FLEAV=F
          GO TO 300
  230 IF(F.GT.FLO.AND.F.LT.FHI) GO TO 210
      PRINT, "OUT OF RANGE IN TRYING TO FIND FLEAV."
      PRINT, "POINT WHERE MAGNITUDE CHARACTERISTIC LEAVES"
      PRINT, "PASSBAND CANNOT BE FOUND BY ROUTINE FINDPK."
      IERR=1
      RETURN
  300 CONTINUE
      FLO=0.
      FHI=FLEAV*3.
C
C     FIND FREQ. OF -90 DEGREE OR WHATEVER PHASE FHLEAV SPECIFIES."
C
      IF(TYPFLAG.EQ.2) GO TO 400
      IF(PEAKING.EQ.0) GO TO 310
          F=FLEAV/10.
          DIR=+1.
          STEP=FO/(Q*20.)
          ERROR=FO*1.0E-7
          DEGPHAS=0.
          GO TO 320
  310 F=FLEAV/10.
      DIR=+1.
      STEP=FO/(Q*20.)
      ERROR=1.0E-7*FO
      DEGPHAS=0.
  320 LASTPHS=DEGPHAS
      THETA=2*PI*F*T
      CALL MAGPHAS(AK,A,B,THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      CROSSED=0
      IF(DEGPHAS.LT.PHLEAV.AND.LASTPHS.GT.PHLEAV) CROSSED=1
      IF(LASTPHS.LT.PHLEAV.AND.DEGPHAS.GT.PHLEAV) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 330
          DIR=-1.*DIR
          STEP=STEP/2.
  330 F=F+STEP*DIR
      IF(STEP.GE.ERROR) GO TO 340
          FPHLEAV=F
          GO TO 400
  340 IF(F.GT.FLO.AND.F.LT.FHI) GO TO 320
      PRINT, "OUT OF RANGE IN TRYING TO FIND FPHLEAV."
      PRINT, "POINT WHERE PHASE CHARACTERISTIC LEAVES"
      PRINT, "PASSBAND CANNOT BE FOUND BY ROUTINE FINDPK."
      IERR=1
      RETURN
  400 CONTINUE
```

```
C
C         NOW FIND REFLECTION ZERO FOR PEAKING FUNCTIONS.
C
      IF(PEAKING.EQ.0) GO TO 500
      F=FFR
      FLO=FFR/2.
      FHI=FFR*2.
      DIR=1.
      STEP=FFR/40.
      ABMAG=0.
      ERROR=FO*1.0E-7
  410 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(AK,A,B,THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      IF(ABMAG.GT.LASTMAG) GO TO 420
          DIR=-1.*DIR
          STEP=STEP/2.
  420 F=F+STEP*DIR
      IF(STEP.GE.ERROR) GO TO 430
          FREFZ=F
          MAGREFZ=ABMAG
          GO TO 500
  430 IF(F.GT.FLO.AND.F.LT.FHI) GO TO 410
          PRINT, "NO REFLECTION ZERO FOUND."
      PRINT, "FINDPK CANNOT FIND REFLECTION ZERO OF"
      PRINT, "CHEBYSHEV FUNCTION."
      IERR=1
          RETURN
  500 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE MWARP(IERR,FP,FO,BW,Q,GO,FC,ALPHA,COUNT,TYPFLAG,
     &AP,EPS,ALPHAR,T,A,B,FREFZ)
      DOUBLE PRECISION ALPHA(8),A(4),B(4)
      DOUBLE PRECISION FP,FO,BW,Q,GO,FC,AP,EPS,ALPHAR
      DOUBLE PRECISION T,FREFZ,DBGO,DLOG10
     INTEGER COUNT,COUNT2,CHOICE,TYPFLAG
C
      DBGO=20.*DLOG10(GO)
C
      PRINT, "ADJUSTS FREQUENCY AND MAGNITUDE TO COMPENSATE"
     PRINT, "FOR Z-TRANSFORM WARPING."
      PRINT,
  10 IF(TYPFLAG.EQ.2) GO TO 100
```

```
          CALL TYPHDR(TYPFLAG)
          PRINT, "YOU MAY CHOOSE TO WARP FO TO APPLICABLE "
          PRINT, "PHASE OR MAGNITUDE."
          PRINT, "ENTER NUMBER OF DESIRED COMMAND."
          PRINT, "     1.  MAGNITUDE"
          PRINT, "     2.  PHASE"
          READ, CHOICE
          CALL MWARP1(FO,BW,DBGO)
          CALL WARPBG(IERR,FO,BW,Q,GO,FC,ALPHA,COUNT,COUNT2,
     &CHOICE,TYPFLAG,T,A,B,ALPHAR)
      IF(IERR.EQ.1) RETURN
          DBGO=20.*DLOG10(GO)
          CALL MWARP2(FO,BW,GO,COUNT,COUNT2)
          RETURN
C
C
  100 CALL TYPHDR(TYPFLAG)
      CALL MWARP1(FO,BW,DBGO)
      PRINT 110, FP
  110 FORMAT("DESIRED PASSBAND EDGE FP= ",G15.8,"  HERTZ")
      PRINT 120, AP
  120 FORMAT("DESIRED RIPPLE WIDTH AP= ",G15.8,"  DB")
      CALL WARPCHB(IERR,FP,AP,FO,BW,Q,GO,FC,ALPHA,COUNT,COUNT2,
     &T,A,B,EPS,ALPHAR,FREFZ)
      IF(IERR.EQ.1) RETURN
      DBGO=20.*DLOG10(GO)
      CALL MWARP2(FO,BW,DBGO,COUNT,COUNT2,FREFZ)
      PRINT 130, AP
  130 FORMAT("CHEBYSHEV RIPPLE WIDTH, AP= ",G15.8,"  DB")
      PRINT 140, FREFZ
  140 FORMAT("REFLECTION ZERO OCCURS AT ",G15.8,"  HERTZ")
      RETURN
      END
C
C
C
C
      SUBROUTINE MWARP1(FO,BW,DBGO)
      DOUBLE PRECISION FO,BW,DBGO
      PRINT 10, FO
   10 FORMAT("DESIRED POLE FREQUENCY, FO= ",G15.8,"  HERTZ")
      PRINT 20, BW
   20 FORMAT("DESIRED POLE BANDWIDTH, BW= ",G15.8,"  HERTZ")
      PRINT 30, DBGO
   30 FORMAT("DESIRED GAIN AT ZERO FREQ., DBGO= ",G15.8,"  DECIBELS")
      RETURN
      END
C
C
C
C
```

```
      SUBROUTINE MWARP2(FO,BW,DBGO,COUNT,COUNT2,FREFZ)
      DOUBLE PRECISION FO,BW,DBGO,FREFZ
      INTEGER COUNT,COUNT2
      PRINT,
      PRINT, "WARPED VARIABLES FOLLOW"
      PRINT 10, FO
   10 FORMAT("FO= ",G15.8,"  HERTZ")
      PRINT 20, BW
   20 FORMAT("BW= ",G15.8,"  HERTZ")
      PRINT 30, DBGO
   30 FORMAT("DBGO= ",G15.8,"  DECIBELS")
      PRINT,
      PRINT 40, COUNT
   40 FORMAT("WARP ITERATION COUNT= ",I4,"  CYCLES")
      PRINT 50, COUNT2
   50 FORMAT("FINDPK ITERATION COUNT= ",I5,"  CYCLES")
      RETURN
      END
```

APPENDIX B

COMPLETE LISTING FOR MARTBP PROGRAM

```
 OLD MARTBP
*LIST

C      MARTBP
C
C
       DOUBLE PRECISION ALPHA(7),KPRIME,C(7),MIN3(2)
       DOUBLE PRECISION CMIN,CSMALL,SUMC,FO,BW,DBGAIN,GAIN
       DOUBLE PRECISION FC,T,A3,A2,A1,A0,B3,B2,B1,B0,ALPHAC,ALPHAR
        DOUBLE PRECISION CPAREA,PFPERSQ
       INTEGER STATE,CMD,COUNT,WARPFLG,CAPFLAG
       DO 10 I=1,7
     10 ALPHA(I)=0.
       DO 20 I=1,7
     20 C(I)=0.
        CMIN=0.
        CSMALL=0.
        SUMC=0.
        FO=0.
        BW=0.
        DBGAIN=0.
        GAIN=0.
        FC=0.
        T=0.
        KPRIME=0.
        A3=0.
        A2=0.
        A1=0.
        A0=0.
        B3=0.
        B2=0.
        B1=0.
        B0=0.
        COUNT=0
        MIN3(1)=0.
        MIN3(2)=0.
        WARPFLG=0
        CAPFLAG=0
C
C
C
```

```
      STATE=0
      CALL HEADER
      PRINT, "NEED MENU? (0=NO, 1=YES)"
      READ, MENUFLG
      IF(MENUFLG.EQ.1) CALL MENU
   90 PRINT,
      PRINT, "ENTER COMMAND NUMBER (#4 FOR MENU)"
      READ, CMD
C
      IERR=0
C
  100 IF(CMD.NE.1) GO TO 200
          CALL ENTFO(FO)
          CALL ENTBW(BW)
          CALL ENTGAIN(DBGAIN,GAIN)
      CALL ENTALC(ALPHAC)
      CALL ENTALR(ALPHAR)
          CALL ENTFC(FC)
          CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,
     &T,ALPHA,A3,A2,A1,A0,B3,B2,B1,B0)
          STATE=1
  200 IF(CMD.NE.2.OR.STATE.EQ.0) GO TO 300
         CALL FRESP(ALPHA,T,A3,A2,A1,A0,B3,B2,B1,B0)
  300 IF(CMD.NE.3.OR.STATE.EQ.0) GO TO 400
          CALL PRECAPS(CMIN,PFPERSQ)
          CALL CAPS(PFPERSQ,CPAREA,ALPHAC,ALPHA,C,CSMALL,SUMC,CMIN)
          CALL POSTCAP(PFPERSQ,CPAREA,CMIN,C,SUMC)
          CAPFLAG=1
  400 IF(CMD.NE.4) GO TO 500
          CALL MENU
  500 IF(CMD.NE.5) GO TO 510
          STOP
  510 IF(CMD.NE.6.OR.STATE.EQ.0) GO TO 520
          CALL MWARP(ALPHAR,IERR,FO,BW,GAIN,
     &DBGAIN,FC,ALPHA,COUNT,MIN3)
          CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
          WARPFLG=1
  520 IF(CMD.NE.7.OR.STATE.LT.1) GO TO 600
          CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
          IF(CAPFLAG.EQ.1) CALL CAPS(PFPERSQ,CPAREA,ALPHAC,
     &ALPHA,C,CSMALL,SUMC,CMIN)
          CALL PRINT(ALPHAC,CPAREA,PFPERSQ,ALPHAR,
     &ALPHA,FO,BW,GAIN,DBGAIN,FC,T,C,CMIN,SUMC,
     &WARPFLG,CAPFLAG,MIN3)
  600 IF(CMD.NE.8.OR.STATE.EQ.0) GO TO 625
          CALL SPECTRM(A3,A2,A1,A0,B3,B2,B1,B0,T)
  625 IF(CMD.NE.9.OR.STATE.EQ.0) GO TO 650
          CALL SWEEP(ALPHAR,IERR,ALPHA,GAIN)
  650 IF(CMD.NE.10.OR.STATE.EQ.0) GO TO 700
```

```
            CALL ENTFO(FO)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
  700 IF(CMD.NE.11.OR.STATE.EQ.0) GO TO 800
            CALL ENTBW(BW)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
  800 IF(CMD.NE.12.OR.STATE.EQ.0) GO TO 900
            CALL ENTGAIN(DBGAIN,GAIN)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
  900 IF(CMD.NE.13.OR.STATE.EQ.0) GO TO 1000
            CALL ENTFC(FC)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
 1000 IF(CMD.NE.14.OR.STATE.EQ.0) GO TO 1100
            CALL ENTALC(ALPHAC)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
 1100 IF(CMD.NE.15.OR.STATE.EQ.0) GO TO 1150
            CALL ENTALR(ALPHAR)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
 1150 IF(CMD.NE.16.OR.STATE.EQ.0) GO TO 1200
            CALL OPTIMIZ(CAPFLAG,ALPHAR,FO,BW,GAIN,FC,
     &T,ALPHA,A3,A2,A1,A0,B3,B2,B1,B0,PFPERSQ,
     &CPAREA,C,CSMALL,SUMC,CMIN,ALPHAC,IERR)
            CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,
     &T,ALPHA,A3,A2,A1,A0,B3,B2,B1,B0)
 1200 IF(STATE.NE.0) GO TO 1275
      PRINT, "YOUR FIRST COMMAND MUST BE #1."
 1275 IF(IERR.EQ.0) GO TO 1300
            CALL ERRMSG
 1300 GO TO 90
      END
C
C
C
C
      SUBROUTINE SPECTRM(A3,A2,A1,A0,B3,B2,B1,B0,T)
      DOUBLE PRECISION FBEG,FEND,FINC,F,TH,THETA,AK
      DOUBLE PRECISION A3,A2,A1,A0,B3,B2,B1,B0
      DOUBLE PRECISION ABMAG,DBMAG,RADPHAS,DEGPHAS,T
      PRINT, "CALCULATES MAGNITUDE AND PHASE AT A SERIES"
      PRINT, "OF FREQUENCY POINTS.  ENTER STARTING AND ENDING"
      PRINT, "FREQUENCIES (HERTZ), SEPARATED BY A COMMA."
    5 READ, FBEG,FEND
      IF(FBEG.GE.0.0D0.AND.FEND.GT.0.0D0) GO TO 7
      PRINT, "INVALID PARAMETER(S), PLEASE REENTER FBEG AND FEND"
      GO TO 5
    7 PRINT, "ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)"
   10 READ, FINC
      IF(FINC.GT.0.0D0) GO TO 20
      PRINT, "INVALID INCREMENT; PLEASE ENTER A POSITIVE VALUE"
      GO TO 10
```

```
 20 PRINT,
    PRINT,
    PRINT 30,
 30 FORMAT(2X,"FREQUENCY (HZ)",4X,"MAGNITUDE (DB)",4X,
   &"MAGNITUDE",8X,"PHASE (DEG)")
    PRINT,
    TH=2.0D0*3.141592653589793*T
    AK=-1.0D0
    F=FBEG
100 IF(F.GT.FEND) GO TO 999
    THETA=TH*F
    CALL MAGPHAS(AK,A3,A2,A1,A0,B3,B2,B1,B0,THETA,
   &ABMAG,DBMAG,RADPHAS,DEGPHAS)
    PRINT 120, F,DBMAG,ABMAG,DEGPHAS
120 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
    F=F+FINC
    GO TO 100
999 RETURN
    END
C
C
C
C
C
    SUBROUTINE ERRMSG
    PRINT, "COMPUTATION HAS BEEN HALTED DUE TO THE"
    PRINT, "OCCURRENCE OF AN ERROR.  CHECK CURRENT"
    PRINT, "PARAMETER VALUES AND RESTART WITH COMMAND #1."
    RETURN
    END
C
C
C
C
    SUBROUTINE SWEEP(ALPHAR,IERR,ALPHA,GAIN)
    DOUBLE PRECISION GAINA,ALPHAA(7),ALPHAR
    DOUBLE PRECISION FCBEG,FCEND,FCINC,TWOPI,K(2),FC,BW
    DOUBLE PRECISION FO,Q,T,GAIN,DSQRT,ALPHA(7)
    DOUBLE PRECISION BWEXP,FPEAK,PHASE,ICOUNT
    DOUBLE PRECISION ICHOICE,PKMAG,MIN3(2)
    PRINT, "ILLUSTRATES EFFECT OF SWEEPING CLOCK FREQUENCIES"
    PRINT, "(FC) ON POLE FREQUENCY AND BANDWIDTH, FOR A"
    PRINT, "SERIES OF CLOCK FREQUENCIES.  ALPHAS AND OTHER"
    PRINT, "PARAMETERS ARE NOT ALTERED BY THIS ROUTINE."
    PRINT, "ENTER BEGINNING AND ENDING CLOCK FREQUENCIES"
    PRINT, "(HERTZ), SEPARATED BY A COMMA."
 10 READ, FCBEG, FCEND
    IF(FCBEG.GT.0.0D0.AND.FCEND.GT.FCBEG) GO TO 20
      PRINT, "ENTER BEGINNING>0 AND END>BEGINNING."
      GO TO 10
 20 PRINT, "ENTER CLOCK FREQUENCY INCREMENT (HERTZ)."
 30 READ, FCINC
    IF(FCINC.GT.0.0D0) GO TO 40
      PRINT, "ENTER INCREMENT>0."
      GO TO 30
```

```
 40 PRINT,
    PRINT,
    PRINT 50,
 50 FORMAT(3X,"CLOCK",15X,"FO",16X,"BW",13X,"PEAKING")
    PRINT 60,
 60 FORMAT(3X,"FREQ.",50X,"FREQ.")
 90 TWOPI=2.0D0*3.141592653589793
    K(1)=ALPHA(3)/TWOPI
    K(2)=ALPHA(4)*DSQRT(ALPHAR)/TWOPI
    DO 300 I=1,7
300 ALPHAA(I)=ALPHA(I)
    GAINA=GAIN
    ALPHARA=ALPHAR
C
C
    FC=FCBEG
100 IF(FC.GT.FCEND) GO TO 999
    BW=K(1)*FC
    FO=K(2)*FC
    Q=FO/BW
    T=1.0D0/FC
    ICOUNT=0
    ICHOICE=1
C
C
C
    CALL FINDPK(ALPHARA,IERR,FO,BW,GAINA,T,FC,ALPHAA,
   &BWEXP,FPEAK,PHASE,ICOUNT,ICHOICE,PKMAG,MIN3)
    IF(IERR.EQ.1) RETURN
190 PRINT 200, FC,FO,BW,FPEAK
200 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
C
C
    FC=FC+FCINC
    GO TO 100
999 RETURN
    END
C
C
C
C
C
    SUBROUTINE OPTIMIZ(CAPFLAG,ALPHAR,FO,BW,GAIN,FC,
   &T,ALPHA,A3,A2,A1,A0,B3,B2,B1,B0,PFPERSQ,
   &CPAREA,C,CSMALL,SUMC,CMIN,ALPHAC,IERR)
    INTEGER CAPFLAG
    DOUBLE PRECISION PARAM,PMIN,PMAX,PINC,PBEST,SUMNEW,SUMBEST
    DOUBLE PRECISION ALPHAR,FO,BW,GAIN,FC,T,ALPHA(7)
    DOUBLE PRECISION A3,A2,A1,A0,B3,B2,B1,B0,PFPERSQ,CPAREA,C(7)
    DOUBLE PRECISION CSMALL,SUMC,CMIN,ALPHAC
C
    IF(CAPFLAG.EQ.1) GO TO 50
      PRINT, "PLEASE CALCULATE CAPACITORS (COMMAND #3) FIRST."
      RETURN
```

```
 50 PRINT, "TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE"
    PRINT, "PARAMETER FROM THE FOLLOWING LIST:"
    PRINT, "      1.  ALPHAC=C2/C1"
    PRINT, "      2.  ALPHAR=ALPHA(7)/ALPHA(4)"
    READ, ICHOICE
    IF(ICHOICE.NE.1.AND.ICHOICE.NE.2) GO TO 50
    PRINT,
 60 PRINT, "ENTER IN ORDER, SEPARATED BY COMMAS:"
    PRINT, "PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT"
    READ, PMIN, PMAX, PINC
    IF(PMIN.GT.0.0D0.AND.PMAX.GT.PMIN.AND.PINC.GT.0.0D0)
   &GO TO 70
    PRINT, "ENTER PMAX>PMIN>0 AND INCREMENT>0."
    GO TO 60
C
C
 70 PARAM=PMIN
    PBEST=PARAM
    SUMBEST=SUMC*1.0D6
C
 80 IF(PARAM.GT.PMAX) GO TO 200
    IF(ICHOICE.EQ.1) ALPHAC=PARAM
    IF(ICHOICE.EQ.2) ALPHAR=PARAM
        CALL ALPHAS(ALPHAR,IERR,FO,BW,GAIN,FC,
   &T,ALPHA,A3,A2,A1,A0,B3,B2,B1,B0)
    IF(IERR.EQ.1) RETURN
    CALL CAPS(PFPERSQ,CPAREA,ALPHAC,ALPHA,C,CSMALL,SUMC,CMIN)
C
    SUMNEW=SUMC
    IF(SUMNEW.GE.SUMBEST) GO TO 100
      PBEST=PARAM
      SUMBEST=SUMNEW
C
100 CONTINUE
    PARAM=PARAM+PINC
C
C
    GO TO 80
C
C
200 CONTINUE
    PRINT, "PARAMETER NAME:"
    IF(ICHOICE.EQ.1) PRINT, "ALPHAC"
    IF(ICHOICE.EQ.2) PRINT, "ALPHAR"
    PRINT 210, PBEST
210 FORMAT("FOR PARAMETER= ",F15.8,"  ,")
    PRINT 220, SUMBEST
220 FORMAT("THE SMALLEST VALUE OF TOTAL CAP. = ",G15.8,"  PF")
C
    IF(ICHOICE.EQ.1) ALPHAC=PBEST
    IF(ICHOICE.EQ.2) ALPHAR=PBEST
C
    PRINT,
    PRINT,
```

```
      PRINT, "THE OPTIMIZED PARAMETER HAS BEEN PLACED"
      PRINT, "IN THE ARGUMENT LIST."
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE MAGPHAS(AK,A3,A2,A1,A0,B3,B2,B1,B0,
     &THETA,ABMG,DBMG,RADPHAS,DEGPHAS)
      DOUBLE PRECISION AK,A3,A2,A1,A0,B3,B2,B1,B0,THETA,ABMG
      DOUBLE PRECISION DBMG,RADPHAS,DEGPHAS,PI,COSTH,COS2TH
      DOUBLE PRECISION COS3TH,SINTH,SIN2TH,SIN3TH,RENUM
      DOUBLE PRECISION AIMNUM,REDEN,AIMDEN,AKPHASE
      DOUBLE PRECISION DCOS,DSIN,DATAN2,DLOG10,DSQRT,DABS
      PI=3.141592653589793
   10 IF(THETA.LT.1.0D3) GO TO 20
          THETA=THETA-PI
          GO TO 10
   20 CONTINUE
      COSTH=DCOS(THETA)
      COS2TH=DCOS(2.0D0*THETA)
      COS3TH=DCOS(3.0D0*THETA)
      SINTH=DSIN(THETA)
      SIN2TH=DSIN(2.0D0*THETA)
      SIN3TH=DSIN(3.0D0*THETA)
      AKPHASE=PI
      IF(AK.GE.0.0D0) AKPHASE=0.0D0
      RENUM=A3*COS3TH+A2*COS2TH+A1*COSTH+A0
      AIMNUM=A3*SIN3TH+A2*SIN2TH+A1*SINTH
      REDEN=B3*COS3TH+B2*COS2TH+B1*COSTH+B0
      AIMDEN=B3*SIN3TH+B2*SIN2TH+B1*SINTH
      ABMG=DSQRT(RENUM**2+AIMNUM**2)/DSQRT(REDEN**2+AIMDEN**2)
      ABMG=ABMG*DABS(AK)
C
C      PI IS ADDED TO RADPHAS TO MAKE ARGUMENT IN
C      THE RANGE 90 TO 270 DEGREES.
C
      RADPHAS=DATAN2(AIMNUM,RENUM)-DATAN2(AIMDEN,REDEN)+PI
      RADPHAS=RADPHAS+AKPHASE
      TWOPI=2.0D0*PI
   30 IF(RADPHAS.LT.TWOPI) GO TO 40
          RADPHAS=RADPHAS-TWOPI
          GO TO 30
   40 CONTINUE
      DBMG=20.0D0*DLOG10(ABMG)
      DEGPHAS=RADPHAS*180.0D0/PI
      RETURN
      END
C
C
```

```
C
      DO 400 I=1,7
      IF(ALPHA(I).GE.0.0D0) GO TO 400
      PRINT, "WARNING:  NEGATIVE ALPHA VALUE--"
      PRINT, "RECHECK INPUT PARAMETERS."
      PRINT 410, I, ALPHA(I)
  410 FORMAT("ALPHA(",I1,") = ",G15.8)
      IERR=1
  400 CONTINUE
      IF(IERR.EQ.1) RETURN
C
C
      RETURN
      END
C
C
C
      SUBROUTINE PRECAPS(CMIN,PFPERSQ)
      DOUBLE PRECISION CMIN,PFPERSQ
          PRINT, "ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF"
          READ, CMIN
      PRINT, "ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL"
      READ, PFPERSQ
      RETURN
      END
C
C
C
      SUBROUTINE CAPS(PFPERSQ,CPAREA,ALPHAC,
     &ALPHA,C,CSMALL,SUMC,CMIN)
      DOUBLE PRECISION ALPHA(7),C(7),CSMALL,SUMC,CMIN,ALPHAC
      DOUBLE PRECISION PFPERSQ,CPAREA
      C(1)=1.
      C(2)=ALPHAC*C(1)
      C(3)=ALPHA(3)*C(1)
      C(4)=ALPHA(4)*C(2)
      C(5)=ALPHA(5)*C(1)
      C(7)=ALPHA(7)*C(1)
C
      CSMALL=DMIN1(C(1),C(2),C(3),C(4),C(5),C(7))
      DO 10 I=1,5
   10 C(I)=C(I)*CMIN/CSMALL
      C(7)=C(7)*CMIN/CSMALL
      SUMC=0.
      DO 30 I=1,5
   30 SUMC=SUMC+C(I)
      SUMC=SUMC+C(7)
C
      CPAREA=SUMC/PFPERSQ
C
C
      RETURN
      END
C
```

```
C
      SUBROUTINE FRESP(ALPHA,T,A3,A2,A1,A0,B3,B2,B1,B0)
       DOUBLE PRECISION ALPHA(7),KPRIME,T,A3,A2,A1,A0
       DOUBLE PRECISION B3,B2,B1.B0,PI,F,THETA,DBMAG
       DOUBLE PRECISION DEGPHAS,ABMAG,RADPHAS
       PI=3.141592653589793
       PRINT. "ENTER EVALUATION FREQUENCY IN HERTZ"
       READ, F
       KPRIME=-1.0D0
       THETA=2*PI*F*T
C
       DBMAG=0.
       DEGPHAS=0.
       CALL MAGPHAS(KPRIME,A3,A2,A1,A0,B3,B2,B1,B0,
      &THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
C
       PRINT 10, F
    10 FORMAT(" AT F= ",G15.8,"  HERTZ")
       PRINT 20, DBMAG
    20 FORMAT(" MAGNITUDE= ",G15.8,"  DB")
       PRINT 30, DEGPHAS
    30 FORMAT("PHASE= ",G15.8,"  DEGREES")
       RETURN
       END
C
C
C
       SUBROUTINE ALPHAS(ALPHAR,IERR,F0,BW,GAIN,FC,T,ALPHA,
      &A3,A2,A1,A0,B3,B2,B1,B0)
       DOUBLE PRECISION ALPHA(7),F0,BW,GAIN,FC,T,TWOPI
       DOUBLE PRECISION A3,A2,A1,A0,B3,B2,B1,B0.ALPHAR,DSQRT
       TWOPI=2.0D0*3.141592653589793
       T=1./FC
C
       ALPHA(3)=TWOPI*BW*T
       ALPHA(4)=(TWOPI*F0*T)/DSQRT(ALPHAR)
       ALPHA(5)=ALPHA(3)*GAIN
       ALPHA(7)=ALPHA(4)*ALPHAR
C
       ALPHA(1)=0.0D0
       ALPHA(2)=0.0D0
       ALPHA(6)=0.0D0
C
       A3=0.
       A2=0.
       A1=-ALPHA(5)
       A0=ALPHA(5)
       B3=0.
       B2=1.
       B1=ALPHA(4)*ALPHA(7)+ALPHA(3)-2.
       B0=1.-ALPHA(3)
C
C
```

```
C
C
      SUBROUTINE POSTCAP(PFPERSQ,CPAREA,CMIN,C,SUMC)
      DOUBLE PRECISION C(7),CMIN,SUMC,PFPERSQ,CPAREA
          PRINT 340, CMIN
340       FORMAT("FOR MINIMUM C= ",G15.8,"  PF")
          PRINT,
          DO 360 I=1,5
          PRINT 370, I,C(I)
370       FORMAT("C(",I1,")= ",G15.8,"  PF")
360       CONTINUE
      I=7
      PRINT 370, I,C(I)
          PRINT,
          PRINT 380, SUMC
380       FORMAT("TOTAL CAPACITANCE USED= ",G15.8,"  PF")
      PRINT 390, PFPERSQ,CPAREA
390   FORMAT("AT ",G11.4," PF/SQ, AREA USED = ",G13.6,"  SQ MILS")
      RETURN
      END
C
C
C
      SUBROUTINE PRINT(ALPHAC,CPAREA,PFPERSQ,
     &ALPHAR,ALPHA,FO,BW,GAIN,DBGAIN,FC,T,C,
     &CMIN,SUMC,WARPFLG,CAPFLAG,MIN3)
      DOUBLE PRECISION ALPHA(7),C(7),MIN3(2),CPAREA,PFPERSQ
      DOUBLE PRECISION FO,BW,GAIN,DBGAIN,FC,T,CMIN
      DOUBLE PRECISION SUMC,ALPHAC,ALPHAR
      INTEGER CAPFLAG,WARPFLG
      PRINT, "VALUES AS CURRENTLY CALCULATED ARE:"
      PRINT 100, FO
100   FORMAT("FO= ",G15.8,"  HERTZ")
      PRINT 110, BW
110   FORMAT("BW= ",G15.8,"  HERTZ")
      PRINT 120, GAIN
120   FORMAT("GAIN= ",G15.8,"  (DIMENSIONLESS)")
      PRINT 130, DBGAIN
130   FORMAT("DBGAIN= ",G15.8,"  DECIBELS")
      PRINT 140, FC
140   FORMAT("FC= ",G15.8,"  HERTZ")
      PRINT 150, T
150   FORMAT("T= ",G15.8,"  SECONDS")
      IF(WARPFLG.EQ.0) GO TO 2
          PRINT, "-3DB POINTS ARE LOCATED AT FREQUENCIES:"
          PRINT 1, MIN3(1),MIN3(2)
  1       FORMAT("F1= ",G15.8,"   AND F2= ",G15.8,"  HERTZ")
          GO TO 4
  2   CONTINUE
  4   DO 10 I=3,5
      PRINT 5, I,ALPHA(I)
  5   FORMAT("ALPHA(",I1,")= ",G15.8)
 10   CONTINUE
```

```
      I=7
      PRINT 5, I, ALPHA(I)
      PRINT 160, ALPHAC
  160 FORMAT("ALPHAC=C(2)/C(1)= ",F15.8)
      PRINT 170, ALPHAR
  170 FORMAT("ALPHAR=ALPHA(7)/ALPHA(4)= ",F15.8)
      IF(CAPFLAG.EQ.1) GO TO 12
          GO TO 30
   12 DO 20 I=1,5
      PRINT 15, I,C(I)
   15 FORMAT("C(",I1,")= ",G15.8,"  PF")
   20 CONTINUE
      I=7
      PRINT 15, I, C(I)
      PRINT 200, CMIN
  200 FORMAT("MINIMUM C= ",G15.8,"  PF")
      PRINT 210, SUMC
  210 FORMAT("TOTAL C= ",G15.8,"  PF")
      PRINT 220, PFPERSQ,CPAREA
  220 FORMAT("AT ",G11.4," PF/SQ, AREA USED = ",G13.6,"  SQ MILS")
   30 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE ENTFO(FO)
      DOUBLE PRECISION FO
      PRINT, "ENTER POLE FREQUENCY, FO (HERTZ)"
      READ, FO
      RETURN
      END
C
C
C
      SUBROUTINE ENTBW(BW)
      DOUBLE PRECISION BW
      PRINT, "ENTER POLE BANDWIDTH, BW (HERTZ)"
      READ, BW
      RETURN
      END
C
C
C
      SUBROUTINE ENTGAIN(DBGAIN,GAIN)
      DOUBLE PRECISION DBGAIN,GAIN
      PRINT, "ENTER PEAK GAIN (DECIBELS)"
      READ, DBGAIN
      GAIN=10.0D0**(DBGAIN/20.0D0)
      RETURN
      END
C
C
C
```

```
      SUBROUTINE ENTALC(ALPHAC)
      DOUBLE PRECISION ALPHAC
      PRINT, "ENTER ALPHAC=C(2)/C(1)"
      READ, ALPHAC
      RETURN
      END
C
C
C
      SUBROUTINE ENTALR(ALPHAR)
      DOUBLE PRECISION ALPHAR
      PRINT, "ENTER ALPHAR=ALPHA(7)/ALPHA(4)"
      READ, ALPHAR
      RETURN
      END
C
C
C
      SUBROUTINE ENTFC(FC)
      DOUBLE PRECISION FC
      PRINT, "ENTER SAMPLING FREQUENCY, FC (HERTZ)"
      READ, FC
      RETURN
      END
C
C
C
      SUBROUTINE HEADER
      PRINT, "SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR"
      PRINT, "MARTIN STATE-VARIABLE BANDPASS FILTER."
      PRINT, "ALGORITHM BY J.A. CONNELLY, FORTRAN VERSION BY "
      PRINT, "W.N. GOOLSBY.  CURRENT REVISION 6/5/81."
      PRINT,
      RETURN
      END
C
C
C
      SUBROUTINE MENU
      PRINT, "A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,"
      PRINT, "ENTER THE NUMBER OF THE APPROPRIATE COMMAND."
      PRINT,
      PRINT,
      PRINT, "1.  ENTER ALL NEW PARAMETERS"
      PRINT, "2.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F"
      PRINT, "3.  FIND CAPACITOR VALUES AND TOTAL C"
      PRINT, "4.  PRINT MENU"
      PRINT, "5.  TERMINATE PROGRAM"
      PRINT, "6.  WARP FO, BW, AND PEAK GAIN."
      PRINT, "7.  PRINT CURRENT PARAMETER VALUES."
      PRINT, "8.  FIND MAGNITUDE AND PHASE AT A SERIES"
      PRINT, "    OF FREQUENCY POINTS."
      PRINT, "9.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON BW & FO."
      PRINT,
```

```
      PRINT, "10.  CHANGE FO ONLY"
      PRINT, "11.  CHANGE BW ONLY"
      PRINT, "12.  CHANGE PEAK GAIN ONLY"
      PRINT, "13.  CHANGE FC ONLY"
      PRINT, "14.  CHANGE ALPHAC=C(2)/C(1) ONLY"
      PRINT, "15.  CHANGE ALPHAR=ALPHA(7)/ALPHA(4) ONLY"
      PRINT, "16.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER"
      PRINT,
      RETURN
      END
C
C
C
C
C
      SUBROUTINE WARP(ALPHAR,IERR,FO,BW,GAIN,FC,ALPHA,COUNT,
     &COUNT2,CHOICE,MIN3)
      DOUBLE PRECISION ALPHA(7),MAGDES,MAGFRAC,MAGERR
      DOUBLE PRECISION FO,BW,GAIN,FC,MIN3(2),MAGDEVN
      DOUBLE PRECISION FDES,BWDES,PHASDES,PI,T,BWERROR
      DOUBLE PRECISION BWFRAC,ERROR,FRAC,BWEXP,FPEAK,PHASE
      DOUBLE PRECISION PKMAG,FDEVN,AFDEVN,BWDEVN,ABWDEVN
      DOUBLE PRECISION AMAGDEV,ALPHAR
      DOUBLE PRECISION DABS
      INTEGER COUNT,COUNT2,RUNFLAG,MAXRUNS,CHOICE
      FDES=FO
      BWDES=BW
      PHASDES=180.0
      MAGDES=GAIN
      PI=3.141592653589793
      T=1./FC
      BWERROR=FDES*1.0E-5
      BWFRAC=1.00
      MAGERR=GAIN*1.0E-4
      MAGFRAC=0.800
      MAXRUNS=50
      COUNT=0
      COUNT2=0
C
C      FIND MAGNITUDE PEAK OR PHASE CROSSING?
C
      IF(CHOICE.EQ.1) GO TO 5
          ERROR=FDES*1.0E-6
          FRAC=1.0
          GO TO 10
    5 ERROR=FDES*1.0E-5
      FRAC=1.000
   10 CALL FINDPK(ALPHAR,IERR,FO,BW,GAIN,T,FC,ALPHA,BWEXP,FPEAK,
     &PHASE,COUNT2,CHOICE,PKMAG,MIN3)
      IF(IERR.EQ.1) RETURN
      RUNFLAG=0
      COUNT=COUNT+1
      FDEVN=FDES-FPEAK
```

```
      AFDEVN=DABS(FDEVN)
      IF(AFDEVN.LT.ERROR) GO TO 20
          FO=FO+FRAC*FDEVN
          RUNFLAG=1
   20 BWDEVN=BWDES-BWEXP
      ABWDEVN=DABS(BWDEVN)
      IF(ABWDEVN.LT.BWERROR) GO TO 30
          BW=BW+BWFRAC*BWDEVN
          RUNFLAG=1
   30 MAGDEVN=MAGDES-PKMAG
      AMAGDEV=DABS(MAGDEVN)
      IF(AMAGDEV.LT.MAGERR) GO TO 35
          GAIN=GAIN+MAGFRAC*MAGDEVN
          RUNFLAG=1
   35 IF(RUNFLAG.EQ.1) GO TO 40
          RETURN
   40 IF(COUNT.LE.MAXRUNS) GO TO 10
          PRINT, "MAXRUNS EXCEEDED IN WARP."
      PRINT, "ITERATION COUNT EXCEEDED WHILE TRYING TO "
      PRINT, "WARP.  WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
          END
C
C
C
      SUBROUTINE FINDPK(ALPHAR,IERR,FO,BW,GAIN,T,FC,ALPHA,BWEXP,
     &FPEAK,PHASE,COUNT,CHOICE,PKMAG,MIN3)
      DOUBLE PRECISION ALPHA(7),LASTMAG,MIN3DB,MIN3(2),ALPHAR
      DOUBLE PRECISION FO,BW,GAIN,T,FC,BWEXP,FPEAK,PHASE
      DOUBLE PRECISION PKMAG,KPRIME,LASTPHS,PI,A3,A2,A1,A0
      DOUBLE PRECISION B3,B2,B1,BO,F,DIVISOR,ABMAG,FHI,FLO
      DOUBLE PRECISION DIR,STEP,ERROR,THETA,DBMAG,RADPHAS
      DOUBLE PRECISION DEGPHAS,PHASDES,FMAGPK,DABS,DSQRT
      INTEGER COUNT,CROSSED,CHOICE
      PI=3.141592653589793
      ITRIAL=0
C
      KPRIME=-1.0D0
      CALL ALPHAS(ALPHAR,IERR,FO,BW.GAIN,FC,T,ALPHA,
     &A3,A2,A1,A0,B3,B2,B1,B0)
      IF(IERR.EQ.1) RETURN
C
      F=FO
      DIVISOR=2.0
      ABMAG=0.
      FHI=FO+BW
      FLO=FO-BW
      DIR=1.
C
C
C      FIND MAGNITUDE PEAK AND RETURN VALUE "PKMAG" AT F.
C
```

```
      STEP=BW/20.
      ERROR=FO*1.0E-6
   10 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A3,A2,A1,A0,B3,B2,B1,B0,
     &THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      IF(ABMAG.GT.LASTMAG) GO TO 20
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
   20 F=F+STEP*DIR
      IF(STEP.GE.ERROR) GO TO 30
          PKMAG=ABMAG
          GO TO 100
   30 IF(F.GT.FLO.AND.F.LT.FHI) GO TO 10
          PRINT, "NO PEAK FOUND BETWEEN FLO AND FHI."
      PRINT, "ROUTINE FINDPK FAILED TO FIND MAGNITUDE PEAK"
      PRINT, "WITHIN PRESCRIBED SEARCH FREQUENCIES.  WARP"
      PRINT, "ATTEMPT ABORTED."
      IERR=1
          RETURN
C
C
C      IF CHOICE=1 THEN NO NEED TO FIND PHASE CROSSING
C
  100 FMAGPK=F
      IF(CHOICE.EQ.1) GO TO 300
C
C
C      FIND PHASE CROSSING FREQUENCY
C
  200 PHASDES=180.0
      F=FO+BW/2.0D0
      DEGPHAS=-400.
      ERROR=FO*1.0E-7
      DIR=-1.0D0
      STEP=BW/40.
  210 LASTPHS=DEGPHAS
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A3,A2,A1,A0,B3,B2,B1,B0,
     &THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      PHASE=DEGPHAS
      COUNT=COUNT+1
      CROSSED=0
      IF(DEGPHAS.LT.PHASDES.AND.LASTPHS.GT.PHASDES) CROSSED=1
      IF(LASTPHS.LT.PHASDES.AND.DEGPHAS.GT.PHASDES) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 220
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
  220 F=F+STEP*DIR
      IF(STEP.LT.ERROR) GO TO 300
      IF(F.GT.FLO.AND.F.LT.FHI) GO TO 210
          PRINT, "OUT OF RANGE IN TRYING TO FIND PHASE CROSSING."
      PRINT, "ROUTINE FINDPK FAILED TO FIND PHASE CROSSING"
      PRINT, "WITHIN PRESCRIBED SEARCH FREQUENCIES.  WARP"
      PRINT, "ATTEMPT ABORTED."
      IERR=1
      RETURN
```

```
C
C
C      FPEAK IS RETURNED AS FREQ. OF MAGNITUDE PEAK
C      FOR CHOICE=1, OR AS FREQ. OF PHASE CROSSING FOR
C      CHOICE=2.
C
  300 FPEAK=F
C
C
C      NOW FIND BANDWIDTH, BW.
C
C
      MIN3DB=PKMAG/DSQRT(2.0D0)
      DO 500 I=1,2
      DIR=(-1.)**I
      F=FMAGPK+DIR*BW/2.0D0
      FHI=FO+BW*3.0D0
      FLO=FO-BW*3.0D0
      STEP=BW/20.
      ERROR=FO*1.0E-6
      ABMAG=PKMAG*2.
  410 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A3,A2,A1,A0,B3,B2,B1,B0,
     &THETA,ABMAG,DBMAG,RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      CROSSED=0
      IF(ABMAG.LT.MIN3DB.AND.LASTMAG.GT.MIN3DB) CROSSED=1
      IF(LASTMAG.LT.MIN3DB.AND.ABMAG.GT.MIN3DB) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 420
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
  420 F=F+STEP*DIR
      IF(STEP.LT.ERROR) GO TO 430
      IF(F.GT.FLO.AND.F.LT.FHI) GO TO 410
          PRINT, "OUT OF RANGE IN TRYING TO FIND 3DB BW."
      PRINT, "ROUTINE FINDPK FAILED TO FIND ONE OR BOTH -3DB"
      PRINT, "POINTS WITHIN THE PRESCRIBED SEARCH FREQUENCIES."
      PRINT, "WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
C
C
  430 MIN3(I)=F
  500 CONTINUE
      BWEXP=DABS(MIN3(2)-MIN3(1))
      RETURN
      END
C
C
```

```
C
      SUBROUTINE MWARP(ALPHAR,IERR,FO,BW,GAIN,
     &DBGAIN,FC,ALPHA,COUNT,MIN3)
      DOUBLE PRECISION ALPHA(7),MIN3(2),FO,BW,GAIN,DBGAIN
      DOUBLE PRECISION FC,DLOG10,ALPHAR
      INTEGER COUNT,COUNT2,CHOICE
      PRINT, "FINDS PREWARPED VALUES FOR FO, BW, AND PEAK GAIN."
      PRINT, "YOU MAY CHOOSE TO WARP FO TO MAGNITUDE PEAK"
      PRINT, "OR TO 180 DEGREE PHASE CROSSING."
      PRINT, "ENTER NUMBER OF DESIRED COMMAND."
      PRINT, "      1.  MAGNITUDE PEAK"
      PRINT, "      2.  180 DEGREE PHASE CROSSING"
      READ, CHOICE
      PRINT 10, FO
   10 FORMAT("DESIRED BANDPASS CENTER= ",G15.8,"HZ")
      PRINT 20, BW
   20 FORMAT("DESIRED BANDWIDTH= ",G15.8,"HZ")
      PRINT 25, DBGAIN
   25 FORMAT("DESIRED PEAK GAIN= ",G15.8,"  DECIBELS")
      PRINT,
      CALL WARP(ALPHAR,IERR,FO,BW,GAIN,FC,ALPHA,
     &COUNT,COUNT2,CHOICE,MIN3)
      IF(IERR.EQ.1) RETURN
      DBGAIN=20.*DLOG10(GAIN)
      PRINT 30, FO
   30 FORMAT("PREWARPED VARIABLES ARE:  FO= ",G15.8,"HZ")
      PRINT 35, BW
   35 FORMAT("BW= ",G15.8,"HZ")
      PRINT 37, DBGAIN
   37 FORMAT("PEAK GAIN= ",G15.8,"  DECIBELS")
      PRINT, "THE -3DB POINTS WERE FOUND AT:"
      PRINT 38, MIN3(1),MIN3(2)
   38 FORMAT("F1= ",G15.8,"   AND F2= ",G15.8,"  HERTZ")
      PRINT,
      PRINT 40, COUNT
   40 FORMAT("WARP ITERATION COUNT= ",I3," CYCLES")
      PRINT 50, COUNT2
   50 FORMAT("FINDPK ITERATION COUNT= ",I5,"  CYCLES.")
      RETURN
      END
```

APPENDIX C

COMPLETE LISTING FOR NOTCH2 PROGRAM

```
C     NOTCH2
C
C
      DOUBLE PRECISION ALPHA(8),C(8),KPRIME,MIN3(2)
      DOUBLE PRECISION CSMALL,SUMC,CMIN,FN,FD,BWN,BWD,RHO
      DOUBLE PRECISION FC,T,A1,A0,B1,B0,THETA,ABMAG,DBMAG
      DOUBLE PRECISION RADPHAS,DEGPHAS,ALPHAC,PFPERSQ,CPAREA
      INTEGER STATE,CMD,COUNT,WARPFLG,CAPFLAG
      DO 10 I=1,8
       C(I)=0.
   10  ALPHA(I)=0.
       SUMC=0.
       CMIN=0.
       FN=0.
       FD=0.
       BWN=0.
       BWD=0.
       RHO=0.
       FC=0.
       T=0.
       KPRIME=0.
       A1=0.
       A0=0.
       B1=0.
       B0=0.
       THETA=0.
       ABMAG=0.
       DBMAG=0.
       RADPHAS=0.
       DEGPHAS=0.
       COUNT=0
       MIN3(1)=0.
       MIN3(2)=0.
       WARPFLG=0
       CAPFLAG=0
       PFPERSQ=1.
       CPAREA=0.
C
C
C
C
C
       STATE=0
       CALL HEADER
       PRINT,
       PRINT, "NEED MENU? (0 =NO, 1 = YES)"
       READ, MENUFLG
       IF(MENUFLG.EQ.1) CALL MENU
   90  PRINT,
       PRINT, "ENTER COMMAND NUMBER (#4 FOR MENU)"
       READ , CMD
```

```
C
      IERR=0
C
  100 IF(CMD.NE.1) GO TO 200
          CALL ENTFN(FN)
          CALL ENTFD(FD)
          CALL ENTBWN(BWN)
          CALL ENTBWD(BWD)
          CALL ENTRHO(RHO)
          CALL ENTFC(FC)
          CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
          STATE=1
          WARPFLG=0
  200 IF(CMD.NE.2.OR.STATE.EQ.0) GO TO 300
          CALL FRESP(ALPHA,T,A1,A0,B1,B0)
  300 IF(CMD.NE.3.OR.STATE.EQ.0) GO TO 400
          PRINT, "ENTER MINIMUM ACCEPTABLE CAPACITANCE IN PF"
          READ, CMIN
          PRINT, "ENTER RATIO C2/C1"
          READ, ALPHAC
          PRINT, "ENTER CHIP CAPACITANCE DENSITY IN PF/SQ MIL"
          READ, PFPERSQ
          CALL CAPS(ALPHA,C,CSMALL,SUMC,CMIN,ALPHAC,PFPERSQ,CPAREA)
          CAPFLAG=1
          PRINT 340, CMIN
  340     FORMAT("FOR MINIMUM C= ",G15.8,"  PF")
          PRINT 350, ALPHAC
  350     FORMAT("AND C2/C1= ",F15.8," ,THE CAPACITANCE VALUES ARE:")
          PRINT,
          DO 360 I=1,8
          PRINT 370, I,C(I)
  370     FORMAT("C(",I1,")= ",G15.8,"  PF")
  360     CONTINUE
          PRINT,
          PRINT 380, SUMC
  380     FORMAT("TOTAL CAPACITANCE USED= ",G15.8,"  PF")
          PRINT 390, PFPERSQ, CPAREA
      390 FORMAT("AT ",G11.4," PF/SQ MIL, AREA USED = ",G13.6,"  SQ MILS")
  400 IF(CMD.NE.4) GO TO 500
          CALL MENU
  500 IF(CMD.NE.5) GO TO 550
          STOP
  550 IF(CMD.NE.6.OR.STATE.EQ.0) GO TO 560
          CALL MWARP(IERR,FN,FD,BWN,BWD,RHO,FC,ALPHA,COUNT,MIN3)
          CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
          WARPFLG=1
  560 IF(CMD.NE.7.OR.STATE.LT.1) GO TO 600
          CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
          IF(CAPFLAG.EQ.1) CALL CAPS(ALPHA,C,CSMALL,SUMC,
     &CMIN,ALPHAC,PFPERSQ,CPAREA)
          CALL PRINT(ALPHAC,ALPHA,FN,FD,BWN,BWD,RHO,FC,T,C,
     &CMIN,SUMC,WARPFLG,CAPFLAG,MIN3,PFPERSQ,CPAREA)
  600 IF(CMD.NE.8.OR.STATE.EQ.0) GO TO 625
          CALL SPECTRM(A1,A0,B1,B0,T)
  625 IF(CMD.NE.9.OR.STATE.EQ.0) GO TO 650
          CALL SWEEP(IERR,ALPHA,RHO)
  650 IF(CMD.NE.10.OR.STATE.EQ.0) GO TO 700
```

```
           CALL ENTFN(FN)
           CALL ALPHAS(IERR,A1.A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
  700 IF(CMD.NE.11.OR.STATE.EQ.0) GO TO 800
           CALL ENTFD(FD)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
  800 IF(CMD.NE.12.OR.STATE.EQ.0) GO TO 900
           CALL ENTBWN(BWN)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
  900 IF(CMD.NE.13.OR.STATE.EQ.0) GO TO 1000
           CALL ENTBWD(BWD)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
 1000 IF(CMD.NE.14.OR.STATE.EQ.0) GO TO 1100
           CALL ENTRHO(RHO)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
 1100 IF(CMD.NE.15.OR.STATE.EQ.0) GO TO 1150
           CALL ENTFC(FC)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
           WARPFLG=0
 1150 IF(CMD.NE.16.OR.STATE.EQ.0) GO TO 1200
      CALL OPTIMIZ(CAPFLAG,IERR,A1,A0,B1,B0,
     &FN,FD,BWN,BWD,RHO,FC,T,ALPHA,C,CSMALL,
     &SUMC,CMIN,ALPHAC,PFPERSQ,CPAREA)
           CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
 1200 IF(STATE.NE.0) GO TO 1275
           PRINT, "YOUR FIRST COMMAND MUST BE #1."
 1275 IF(IERR.EQ.0) GO TO 1300
           CALL ERRMSG
 1300 GO TO 90
      END

C
C
C
C
C
      SUBROUTINE SPECTRM(A1,A0,B1,B0,T)
      DOUBLE PRECISION FBEG,FEND,FINC,F,TH,THETA,AK
      DOUBLE PRECISION A1,A0,B1,B0
      DOUBLE PRECISION ABMAG,DBMAG,RADPHAS,DEGPHAS,T
      PRINT, "CALCULATES MAGNITUDE AND PHASE AT A SERIES"
      PRINT, "OF FREQUENCY POINTS.  ENTER STARTING AND ENDING"
      PRINT, "FREQUENCIES (HERTZ), SEPARATED BY A COMMA."
    5 READ, FBEG,FEND
      IF(FBEG.GE.0.0D0.AND.FEND.GT.0.0D0) GO TO 7
      PRINT, "INVALID PARAMETER(S), PLEASE REENTER FBEG AND FEND"
      GO TO 5
    7 PRINT, "ENTER FREQUENCY INCREMENT BETWEEN POINTS (HERTZ)"
   10 READ, FINC
      IF(FINC.GT.0.0D0) GO TO 20
      PRINT, "INVALID INCREMENT; PLEASE ENTER A POSITIVE VALUE"
      GO TO 10
   20 PRINT,
```

```
      PRINT,
      PRINT 30,
   30 FORMAT(2X,"FREQUENCY (HZ)",4X,"MAGNITUDE (DB)",4X,
     &"MAGNITUDE",9X,"PHASE (DEG)")
      PRINT,
      TH=2.0D0*3.141592653589793*T
      AK=B0
      F=FBEG
  100 IF(F.GT.FEND) GO TO 999
      THETA=TH*F
      CALL MAGPHAS(AK,A1,A0,B1,B0,THETA,ABMAG,DBMAG,
     &RADPHAS,DEGPHAS)
      PRINT 120, F,DBMAG,ABMAG,DEGPHAS
  120 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
      F=F+FINC
      GO TO 100
  999 RETURN
      END
C
C
C
C
C
      SUBROUTINE ERRMSG
      PRINT, "COMPUTATION HAS BEEN HALTED DUE TO THE"
      PRINT, "OCCURRENCE OF AN ERROR.  CHECK CURRENT"
      PRINT, "PARAMETER VALUES AND RESTART WITH COMMAND #1."
      RETURN
      END
C
C
C
C
      SUBROUTINE SWEEP(IERR,ALPHA,RHO)
      DOUBLE PRECISION FCBEG,FCEND,FCINC,TWOPI,K(4),FC,BWEXP
      DOUBLE PRECISION FN,FD,BWN,BWD,Q,T,DSQRT,ALPHA(8)
      DOUBLE PRECISION RHO,FPEAK,PHASE,MIN3(2),SQRHO
      DOUBLE PRECISION ALPHAA(8),SQRHOA
      PRINT, "ILLUSTRATES EFFECT OF SWEEPING CLOCK FREQUENCIES"
      PRINT, "(FC) ON POLE FREQUENCY AND BANDWIDTH, FOR A"
      PRINT, "SERIES OF CLOCK FREQUENCIES.  ALPHAS AND OTHER"
      PRINT, "PARAMETERS ARE NOT ALTERED BY THIS ROUTINE."
      PRINT, "ENTER BEGINNING AND ENDING CLOCK FREQUENCIES"
      PRINT, "(HERTZ), SEPARATED BY A COMMA."
   10 READ, FCBEG, FCEND
      IF(FCBEG.GT.0.0D0.AND.FCEND.GT.FCBEG) GO TO 20
        PRINT, "ENTER BEGINNING>0 AND END>BEGINNING."
        GO TO 10
   20 PRINT, "ENTER CLOCK FREQUENCY INCREMENT (HERTZ)."
   30 READ, FCINC
      IF(FCINC.GT.0.0D0) GO TO 40
        PRINT, "ENTER INCREMENT>0."
        GO TO 30
   40 PRINT,
      PRINT,
```

```
      PRINT 50,
   50 FORMAT(3X,"CLOCK",15X,"FN",16X,"BWD",13X,"NOTCH")
      PRINT 60,
   60 FORMAT(3X,"FREQ.",50X,"CENTER")
   90 TWOPI=2.0D0*3.141592653589793
      SQRHO=DSQRT(RHO)
      K(1)=ALPHA(5)/TWOPI
      K(2)=ALPHA(6)*SQRHO/TWOPI
      K(3)=ALPHA(4)*SQRHO/ALPHA(3)
      K(4)=DSQRT((ALPHA(8)*SQRHO)/(TWOPI*ALPHA(3)))
C
      DO 300 I=1,8
  300 ALPHAA(I)=ALPHA(I)
      SQRHOA=DSQRT(RHO)
C
C
      FC=FCBEG
  100 IF(FC.GT.FCEND) GO TO 999
      BWD=K(1)*FC
      FD=K(2)*FC
      BWN=K(3)*FD
      FN=K(4)*DSQRT(FD*FC)
      T=1.0D0/FC
      ICOUNT=0
      ICHOICE=1
C
C
C
      CALL FINDPK(IERR,FN,BWD,T,ALPHAA,SQRHOA,FD,
     &BWEXP,FPEAK,PHASE,ICOUNT,ICHOICE,MIN3)
      IF(IERR.EQ.1) RETURN
  190 PRINT 200, FC,FN,BWD,FPEAK
  200 FORMAT(G15.8,3X,G15.8,3X,G15.8,3X,G15.8)
C
C
      FC=FC+FCINC
      GO TO 100
  999 RETURN
      END
C
C
C
C
C
      SUBROUTINE OPTIMIZ(CAPFLAG,IERR,A1,A0,B1,B0,
     &FN,FD,BWN,BWD,RHO,FC,T,ALPHA,C,CSMALL,
     &SUMC,CMIN,ALPHAC,PFPERSQ,CPAREA)
      INTEGER CAPFLAG
      DOUBLE PRECISION PARAM,PMIN,PMAX,PINC,PBEST,SUMNEW,SUMBEST
      DOUBLE PRECISION A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC
      DOUBLE PRECISION T,ALPHA(8),C(8),CSMALL,SUMC,CMIN
      DOUBLE PRECISION ALPHAC,PFPERSQ,CPAREA
```

```
C
      IF(CAPFLAG.EQ.1) GO TO 50
        PRINT, "PLEASE CALCULATE CAPACITORS (COMMAND #3) FIRST."
        RETURN
   50 PRINT, "TO MINIMIZE TOTAL CAPACITANCE,CHOOSE ONE"
      PRINT, "PARAMETER FROM THE FOLLOWING LIST:"
      PRINT, "      1.  ALPHAC=C2/C1"
      PRINT, "      2.  RHO"
      READ, ICHOICE
      IF(ICHOICE.NE.1.AND.ICHOICE.NE.2) GO TO 50
      PRINT,
   60 PRINT, "ENTER IN ORDER, SEPARATED BY COMMAS:"
      PRINT, "PARAMETER MIN. VALUE, MAX. VALUE, STEP INCREMENT"
      READ, PMIN, PMAX, PINC
      IF(PMIN.GT.0.0D0.AND.PMAX.GT.PMIN.AND.PINC.GT.0.0D0)
     &GO TO 70
      PRINT, "ENTER PMAX>PMIN>0 AND INCREMENT>0."
      GO TO 60
C
C
   70 PARAM=PMIN
      PBEST=PARAM
      SUMBEST=SUMC*1.0D6
C
   80 IF(PARAM.GT.PMAX) GO TO 200
      IF(ICHOICE.EQ.1) ALPHAC=PARAM
      IF(ICHOICE.EQ.2) RHO=PARAM
      CALL ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
      IF(IERR.EQ.1) RETURN
      CALL CAPS(ALPHA,C,CSMALL,SUMC,CMIN,ALPHAC,PFPERSQ,CPAREA)
C
      SUMNEW=SUMC
      IF(SUMNEW.GE.SUMBEST) GO TO 100
        PBEST=PARAM
        SUMBEST=SUMNEW
C
  100 CONTINUE
      PARAM=PARAM+PINC
C
C
      GO TO 80
C
C
  200 CONTINUE
      PRINT, "PARAMETER NAME:"
      IF(ICHOICE.EQ.1) PRINT, "ALPHAC"
      IF(ICHOICE.EQ.2) PRINT, "RHO"
      PRINT 210, PBEST
  210 FORMAT("FOR PARAMETER= ",F15.8,"  ,")
      PRINT 220, SUMBEST
  220 FORMAT("THE SMALLEST VALUE OF TOTAL CAP. = ",G15.8,"  PF")
C
      IF(ICHOICE.EQ.1) ALPHAC=PBEST
      IF(ICHOICE.EQ.2) RHO=PBEST
C
```

```
      PRINT,
      PRINT,
      PRINT, "THE OPTIMIZED PARAMETER HAS BEEN PLACED"
      PRINT, "IN THE ARGUMENT LIST."
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE MAGPHAS(AK,A1,A0,B1,B0,THETA,
     &ABMG,DBMG,RADPHAS,DEGPHAS)
      DOUBLE PRECISION AK,A1,A0,B1,B0,THETA,ABMG,DBMG
      DOUBLE PRECISION RADPHAS,DEGPHAS,AKPHASE,RENUM
      DOUBLE PRECISION AIMNUM,REDEN,AIMDEN,DSQRT,DSIN,DCOS
      DOUBLE PRECISION DABS,DATAN2,DLOG10
      DOUBLE PRECISION PI,COSTH,COS2TH,SINTH,SIN2TH
      PI=3.141592653589793
   10 IF(THETA.LT.3.0D3) GO TO 20
          THETA=THETA-PI
          GO TO 10
   20 CONTINUE
      COSTH=DCOS(THETA)
      COS2TH=DCOS(2.0D0*THETA)
      SINTH=DSIN(THETA)
      SIN2TH=DSIN(2.0D0*THETA)
      RENUM=COS2TH+A1*COSTH+A0
      AIMNUM=SIN2TH+A1*SINTH
      REDEN=COS2TH+B1*COSTH+B0
      AIMDEN=SIN2TH+B1*SINTH
      ABMG=DSQRT(RENUM**2+AIMNUM**2)/DSQRT(REDEN**2+AIMDEN**2)
      ABMG=ABMG*DABS(AK)
      DBMG=20.0D0*DLOG10(ABMG)
      AKPHASE=PI
      IF(AK.GE.0.0D0) AKPHASE=0.0D0
      RADPHAS=DATAN2(AIMNUM,RENUM)-DATAN2(AIMDEN,REDEN)+AKPHASE
C     ADD PI TO RADPHAS TO GET PHASE IN THE RANGE 0<PHASE<360 DEGREES.
      RADPHAS=RADPHAS+PI
      DEGPHAS=RADPHAS*180./PI
      RETURN
      END
C
C
C
C
C
      SUBROUTINE FRESP(ALPHA,T,A1,A0,B1,B0)
      DOUBLE PRECISION ALPHA(8),KPRIME,T,A1,A0,B1,B0,PI,F
      DOUBLE PRECISION THETA,DBMAG,DEGPHASE
      PI=3.141592653589793
      PRINT, "ENTER EVALUATION FREQUENCY IN HERTZ"
      READ, F
      KPRIME=1./(1.+ALPHA(5))
      THETA=2*PI*F*T
```

```
C
      DBMAG=0.
      DEGPHAS=0.
      CALL MAGPHAS(KPRIME,A1,A0,B1,B0,THETA,0.0D0,DBMAG,0.0D0,
     &DEGPHAS)
C
      PRINT 10, F
   10 FORMAT(" AT F= ",G15.8,"  HERTZ")
      PRINT 20, DBMAG
   20 FORMAT(" MAGNITUDE= ",G15.8,"  DB")
      PRINT 30, DEGPHAS
   30 FORMAT("PHASE= ",G15.8,"  DEGREES")
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ALPHAS(IERR,A1,A0,B1,B0,FN,FD,BWN,BWD,RHO,FC,T,ALPHA)
      DOUBLE PRECISION ALPHA(8),A1,A0,B1,B0,FN,FD,BWN,BWD
      DOUBLE PRECISION RHO,FC,T,ALPHA,TWOPI,SQRHO,DSQRT
      TWOPI=2.0D0*3.141592653589793
      T=1./FC
      ALPHA(3)=1.
      SQRHO=DSQRT(RHO)
      ALPHA(4)=(BWN*ALPHA(3))/(FD*SQRHO)
      ALPHA(5)=TWOPI*BWD*T
      ALPHA(6)=TWOPI*FD*T/SQRHO
      ALPHA(7)=RHO*ALPHA(6)
      ALPHA(8)=(TWOPI*T*ALPHA(3)*FN*FN)/(FD*SQRHO)
      ALPHA(1)=0.
      ALPHA(2)=0.
      A1=ALPHA(4)*ALPHA(7)*(ALPHA(8)/ALPHA(4)+1.)-2.
      A0=1.-ALPHA(7)*ALPHA(4)
      B1=-(2.+ALPHA(5)-ALPHA(6)*ALPHA(7))/(1.+ALPHA(5))
      B0=1./(1.+ALPHA(5))
C
C
      DO 400 I=3,8
      IF(ALPHA(I).GE.0.0D0) GO TO 400
      PRINT, "WARNING:  NEGATIVE ALPHA VALUE--"
      PRINT, "RECHECK INPUT PARAMETERS."
      PRINT 410, I, ALPHA(I)
  410 FORMAT("ALPHA(",I1,") = ",G15.8)
      IERR=1
  400 CONTINUE
      IF(IERR.EQ.1) RETURN
C
C
      RETURN
      END
C
C
C
C
```

```
C
      SUBROUTINE CAPS(ALPHA,C,CSMALL,SUMC,CMIN,ALPHAC,PFPERSQ,CPAREA)
      DOUBLE PRECISION ALPHA(8),CSMALL,SUMC,CMIN,ALPHAC,C(8)
      DOUBLE PRECISION DMIN1,PFPERSQ,CPAREA
C
C
      C(1)=1.
      C(2)=ALPHAC
      C(3)=1.
      C(4)=ALPHA(4)*C(2)
      C(5)=ALPHA(5)
      C(6)=ALPHA(6)*C(2)
      C(7)=ALPHA(7)
      C(8)=ALPHA(8)*C(2)
C
C
      CSMALL=DMIN1(C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8))
      DO 10 I=1,8
   10 C(I)=C(I)*CMIN/CSMALL
      SUMC=0.
      DO 30 I=1,8
   30 SUMC=SUMC+C(I)
      CPAREA=SUMC/PFPERSQ
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PRINT(ALPHAC,ALPHA,FN,FD,BWN,BWD,RHO,FC,T,C,
     &CMIN,SUMC,WARPFLG,CAPFLAG,MIN3,PFPERSQ,CPAREA)
      DOUBLE PRECISION ALPHAC,ALPHA(8),C(8),MIN3(2)
      DOUBLE PRECISION FN,FD,BWN,BWD,RHO,FC,T,CMIN,SUMC
      DOUBLE PRECISION PFPERSQ,CPAREA
      INTEGER CAPFLAG,WARPFLG
      PRINT, "VALUES AS CURRENTLY CALCULATED ARE:"
      PRINT 100, FN
  100 FORMAT("FN= ",G15.8,"  HERTZ")
      PRINT 110, FD
  110 FORMAT("FD= ",G15.8,"  HERTZ")
      PRINT 120, BWN
  120 FORMAT("BWN= ",G15.8,"  HERTZ")
      PRINT 130, BWD
  130 FORMAT("BWD= ",G15.8,"  HERTZ")
      PRINT 140, RHO
  140 FORMAT("RHO= ",F15.8,"  (DIMENSIONLESS)")
      PRINT 145, ALPHAC
  145 FORMAT("ALPHAC= ",F15.8,"  (DIMENSIONLESS)")
      PRINT 150, FC
  150 FORMAT("FC= ",G15.8,"  HERTZ")
      PRINT 160, T
  160 FORMAT("T= ",G15.8,"  SECONDS")
      IF(WARPFLG.EQ.0) GO TO 2
          PRINT, "-3DB POINTS ARE LOCATED AT FREQUENCIES:"
          PRINT 1, MIN3(1),MIN3(2)
    1     FORMAT("F1= ",G15.8,"   AND F2= ",G15.8,"  HERTZ")
```

```
          GO TO 4
    2 CONTINUE
    4 DO 10 I=3,8
      PRINT 5, I,ALPHA(I)
    5 FORMAT("ALPHA(",I1,")= ",G15.8)
   10 CONTINUE
      IF(CAPFLAG.EQ.1) GO TO 12
          GO TO 30
   12 DO 20 I=1,8
      PRINT 15, I,C(I)
   15 FORMAT("C(",I1,")= ",G15.8,"  PF")
   20 CONTINUE
      PRINT 22, CMIN
   22 FORMAT("MINIMUM C = ",G15.8,"  PF")
      PRINT 23, SUMC
   23 FORMAT("TOTAL C = ",G15.8,"  PF")
      PRINT 25, PFPERSQ, CPAREA
   25 FORMAT("AT ",G11.4," PF/SQ MIL, AREA USED = ",
     &G13.6,"  SQ MILS")
   30 CONTINUE
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTFN(FN)
      DOUBLE PRECISION FN
      PRINT, "ENTER FN (HERTZ)"
      READ, FN
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTFD(FD)
      DOUBLE PRECISION FD
      PRINT, "ENTER FD (HERTZ)"
      READ, FD
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTBWN(BWN)
      DOUBLE PRECISION BWN
      PRINT, "ENTER BWN (HERTZ)"
      READ, BWN
      RETURN
      END
C
```

```
C
C
C
C
      SUBROUTINE ENTBWD(BWD)
      DOUBLE PRECISION BWD
      PRINT, "ENTER BWD (HERTZ)"
      READ, BWD
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTRHO(RHO)
      DOUBLE PRECISION RHO
      PRINT, "ENTER RHO"
      READ, RHO
      RETURN
      END
C
C
C
C
C
      SUBROUTINE ENTFC(FC)
      DOUBLE PRECISION FC
      PRINT, "ENTER FC (HERTZ)"
      READ, FC
      RETURN
      END
C
C
C
C
C
      SUBROUTINE HEADER
      PRINT, "SWITCHED-CAPACITOR FILTER DESIGN PROGRAM FOR"
      PRINT, "TIMCON NOTCH FILTER. ALGORITHM BY J.A. CONNELLY,"
      PRINT, "FORTRAN VERSION BY W.N. GOOLSBY."
      PRINT, "REVISION 6/9/81."
      PRINT,
      RETURN
      END
C
C
C
C
```

```
C
      SUBROUTINE MENU
      PRINT, "A MENU OF COMMANDS FOLLOWS.  TO EXECUTE A FUNCTION,"
      PRINT, "ENTER THE NUMBER OF THE APPROPRIATE COMMAND."
      PRINT,
      PRINT,
      PRINT, "1.  ENTER ALL NEW PARAMETERS"
      PRINT, "2.  CALCULATE MAGNITUDE AND PHASE AT FREQ.=F"
      PRINT, "3.  FIND CAPACITOR VALUES AND TOTAL C"
      PRINT, "4.  PRINT MENU"
      PRINT, "5.  TERMINATE PROGRAM"
      PRINT, "6.  WARP FN AND BWD."
      PRINT, "7.  PRINT CURRENT PARAMETER VALUES."
      PRINT, "8.  FIND MAGNITUDE AND PHASE AT A SERIES"
      PRINT, "    OF FREQUENCY POINTS."
      PRINT, "9.  VIEW EFFECT OF SWEEPING CLOCK FREQ. ON FN & BWD"
      PRINT, "10.  CHANGE FN ONLY"
      PRINT, "11.  CHANGE FD ONLY"
      PRINT, "12.  CHANGE BWN ONLY"
      PRINT, "13.  CHANGE BWD ONLY"
      PRINT, "14.  CHANGE RHO ONLY"
      PRINT, "15.  CHANGE FC ONLY"
      PRINT, "16.  MINIMIZE TOTAL CAP. BY VARYING ONE PARAMETER"
      PRINT,
      RETURN
      END
C
C
C
C
C
      SUBROUTINE WARP(IERR,FN,FD,BWN,BWD,RHO,FC,ALPHA,COUNT,
     &COUNT2,CHOICE,MIN3)
      DOUBLE PRECISION ALPHA(8),MIN3(2),FN,FD,BWN,BWD,RHO
      DOUBLE PRECISION FC,FDES,BWDES,PHASDES,PI,T,SQRHO,BWERROR
      DOUBLE PRECISION ERROR,FRAC,BW,FPEAK,PHASE,FDEVN,AFDEVN
      DOUBLE PRECISION BWFRAC,BWDEVN,ABWDEVN,DSQRT,DABS
      INTEGER COUNT,COUNT2,RUNFLAG,MAXRUNS,CHOICE
      FDES=FN
      BWDES=BWD
      PHASDES=180.0
      PI=3.141592653589793
      T=1./FC
      ALPHA(1)=0.
      ALPHA(2)=0.
      ALPHA(3)=1.
      SQRHO=DSQRT(RHO)
      ALPHA(4)=(BWN*ALPHA(3))/(FD*SQRHO)
      ALPHA(6)=2*PI*FD*T/SQRHO
      ALPHA(7)=RHO*ALPHA(6)
      BWERROR=FDES*1.0E-5
      BWFRAC=1.00
      MAXRUNS=50
      COUNT=0
      COUNT2=0
C
```

```
C     FIND MAG. PEAK OR PHASE CROSSING?
C
      IF(CHOICE.EQ.1) GO TO 5
          ERROR=FDES*1.0E-6
          FRAC=1.0
          GO TO 10
    5 ERROR=FDES*1.0E-5
      FRAC=1.000
   10 CALL FINDPK(IERR,FN.BWD,T,ALPHA,SQRHO,FD.BW,FPEAK,PHASE,
     &COUNT2,CHOICE,MIN3)
      IF(IERR.EQ.1) RETURN
      RUNFLAG=0
      COUNT=COUNT+1
      FDEVN=FDES-FPEAK
      AFDEVN=DABS(FDEVN)
      IF(AFDEVN.LT.ERROR) GO TO 20
          FN=FN+FRAC*FDEVN
          RUNFLAG=1
   20 BWDEVN=BWDES-BW
      ABWDEVN=DABS(BWDEVN)
      IF(ABWDEVN.LT.BWERROR) GO TO 30
          BWD=BWD+BWFRAC*BWDEVN
          RUNFLAG=1
   30 IF(RUNFLAG.EQ.1) GO TO 40
          RETURN
   40 IF(COUNT.LE.MAXRUNS) GO TO 10
          PRINT, "MAXRUNS EXCEEDED IN WARP."
      PRINT, "MAXIMUM ITERATION COUNT EXCEEDED IN ROUTINE"
      PRINT, "WARP.  WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
          END
C
C
C
C
C
      SUBROUTINE FINDPK(IERR,FN,BWD,T,ALPHA,SQRHO,
     &FD,BW,FPEAK,PHASE,COUNT,CHOICE,MIN3)
      DOUBLE PRECISION ALPHA(8),LASTMAG,MIN3DB,MIN3(2),KPRIME
      DOUBLE PRECISION FN,BWD,T,SQRHO,FD,BW,FPEAK,PHASE,LASTPHS
      DOUBLE PRECISION PI,A1,A0,B1,B0,F,DIVISOR,ABMAG,FHI,FLO
      DOUBLE PRECISION DIR,STEP,ERROR,THETA,DBMAG,RADPHAS
      DOUBLE PRECISION DEGPHAS,PHASDES,DABS
      INTEGER COUNT,CROSSED,CHOICE
      PI=3.141592653589793
      ALPHA(5)=2*PI*BWD*T
      ALPHA(8)=(2*PI*T*ALPHA(3)*FN*FN)/(FD*SQRHO)
      KPRIME=1./(1.+ALPHA(5))
      A1=ALPHA(4)*ALPHA(7)*(ALPHA(8)/ALPHA(4)+1.)-2.
      A0=1.-ALPHA(7)*ALPHA(4)
      B1=-(2.+ALPHA(5)-ALPHA(6)*ALPHA(7))/(1.+ALPHA(5))
      B0=1./(1.+ALPHA(5))
C
```

```
C
      F=FN
      DIVISOR=2.0
      ABMAG=0.
      FHI=FN+BWD/2.
      FLO=FN-BWD/2.
      DIR=1.
C
C     CHOOSE TO FIND MAG. PEAK OR 180 DEGREE CROSSING.
C
      IF(CHOICE.EQ.2) GO TO 200
C
      STEP=BWD/20.
      ERROR=FN*1.0E-6
   10 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A1,A0,B1,B0,THETA,ABMAG,DBMAG,
     &RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      IF(ABMAG.LT.LASTMAG) GO TO 20
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
   20 F=F+STEP*DIR
      IF(STEP.LT.ERROR) GO TO 300
      IF(F.GT.FLO.AND.F.LT.FHI) GO TO 10
          PRINT, "NO PEAK FOUND BETWEEN FLO AND FHI."
      PRINT, "ROUTINE FINDPK FAILED TO FIND MAGNITUDE MINIMUM"
      PRINT, "WITHIN PRESCRIBED SEARCH FREQUENCIES."
      PRINT, "WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
C
C
C
  200 PHASDES=180.0
      DEGPHAS=-1.0
      ERROR=FN*1.0E-7
      STEP=BWD/40.
  210 LASTPHS=DEGPHAS
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A1,A0,B1,B0,THETA,ABMAG,DBMAG,
     &RADPHAS,DEGPHAS)
      PHASE=DEGPHAS
      COUNT=COUNT+1
      CROSSED=0
      IF(DEGPHAS.LT.PHASDES.AND.LASTPHS.GT.PHASDES) CROSSED=1
      IF(LASTPHS.LT.PHASDES.AND.DEGPHAS.GT.PHASDES) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 220
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
  220 F=F+STEP*DIR
      IF(STEP.LT.ERROR) GO TO 300
```

```
      IF(F.GT.FLO.AND.F.LT.FHI) GO TO 210
          PRINT, "OUT OF RANGE IN TRYING TO FIND PHASE CROSSING."
      PRINT, "ROUTINE FINDPK FAILED TO FIND PHASE CROSSING"
      PRINT, "WITHIN PRESCRIBED SEARCH FREQUENCIES."
      PRINT, "WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
C
C
C
C
C
  300 FPEAK=F
C
C
C     NOW FIND BANDWIDTH, BW.
C
C
      MIN3DB=1./DSQRT(2.)
      DO 500 I=1,2
      DIR=(-1.)**I
      F=FPEAK+DIR*BWD/2.
      FHI=FN+BWD
      FLO=FN-BWD
      STEP=BWD/20.
      ERROR=FN*1.0E-6
      ABMAG=0.
  410 LASTMAG=ABMAG
      THETA=2*PI*F*T
      CALL MAGPHAS(KPRIME,A1,A0,B1,B0,THETA,ABMAG,DBMAG,
     &RADPHAS,DEGPHAS)
      COUNT=COUNT+1
      CROSSED=0
      IF(ABMAG.LT.MIN3DB.AND.LASTMAG.GT.MIN3DB) CROSSED=1
      IF(LASTMAG.LT.MIN3DB.AND.ABMAG.GT.MIN3DB) CROSSED=1
      IF(CROSSED.EQ.0) GO TO 420
          DIR=-1.*DIR
          STEP=STEP/DIVISOR
  420 F=F+STEP*DIR
      IF(STEP.LT.ERROR) GO TO 430
      IF(F.GT.FLO.AND.F.LT.FHI) GO TO 410
          PRINT, "OUT OF RANGE IN TRYING TO FIND 3DB BW."
      PRINT, "ROUTINE FINDPK FAILED TO FIND ONE OR BOTH"
      PRINT, "-3DB POINTS WITHIN PRECSRIBED SEARCH"
      PRINT, "FREQUENCIES.  WARP ATTEMPT ABORTED."
      IERR=1
          RETURN
  430 MIN3(I)=F
  500 CONTINUE
      BW=DABS(MIN3(2)-MIN3(1))
      RETURN
      END
C
C
C
```

```
C
C
      SUBROUTINE MWARP(IERR,FN,FD,BWN,BWD,RHO,FC,ALPHA,COUNT,MIN3)
      DOUBLE PRECISION ALPHA(8),MIN3(2)
      DOUBLE PRECISION FN,FD,BWN,BWD,RHO,FC
      INTEGER COUNT,COUNT2,CHOICE
      PRINT, "FINDS PREWARPED VALUES FOR FN AND BWD."
      PRINT, "YOU MAY CHOUSE TO WARP FN TO MAGNITUDE PEAK"
      PRINT, "OR TO 180 DEGREE PHASE CROSSING."
      PRINT, "ENTER NUMBER OF DESIRED COMMAND."
      PRINT, "      1.  MAGNITUDE PEAK"
      PRINT, "      2.  180 DEGREE PHASE CROSSING"
      READ, CHOICE
      PRINT 10, FN
   10 FORMAT("DESIRED NOTCH CENTER= ",G15.8,"HZ")
      PRINT 20, BWD
   20 FORMAT("DESIRED BANDWIDTH= ",G15.8,"HZ")
      CALL WARP(IERR,FN,FD,BWN,BWD,RHO,FC,ALPHA,COUNT,
     &COUNT2,CHOICE,MIN3)
      IF(IERR.EQ.1) RETURN
      PRINT 30, FN
   30 FORMAT("PREWARPED VARIABLES ARE:  FN= ",G15.8,"HZ")
      PRINT 35, BWD
   35 FORMAT("BWD= ",G15.8,"HZ")
      PRINT, "-3DB POINTS WERE FOUND AT:"
      PRINT 37, MIN3(1),MIN3(2)
   37 FORMAT("F1= ",G15.8,"   AND F2= ",G15.8,"  HERTZ")
      PRINT 40, COUNT
   40 FORMAT("WARP ITERATION COUNT= ",I3," CYCLES")
      PRINT 50, COUNT2
   50 FORMAT("FINDPK ITERATION COUNT= ",I5,"  CYCLES.")
      RETURN
      END
```